NON-LINEAR ADAPTIVE EQUALIZATION

BASED ON

A MULTI-LAYER PERCEPTRON ARCHITECTURE

Sammy Siu (Sze Kee Siu) B.Sc., M.Sc., A.M.I.E.E.

Thesis Submitted for the degree

of Doctor of Philosophy to the

Faculty of Science

University of Edinburgh

1990

Abstract

The subject of this thesis is the original study of the application of the multi-layer perceptron architecture to channel equalization in digital communications systems. Both theoretical analyses and simulations were performed to explore the performance of the perceptron-based equalizer (including the decision feedback equalizer). Topics covered include the factors that affect performance of the structures including, the parameters (learning gain and momentum parameter) in the learning algorithm, the network topology (input dimension, number of neurons and the number of hidden layers), and the power metrics on the error cost function. Based on the geometric hyperplane analysis of the multi-layer perceptron, the results offer valuable insight into the properties and complexity of the network. Comparisons of the bit error rate performance and the dynamic behaviour of the decision boundary of the perceptron-based equalizer with both the optimal non-linear equalizer and the optimal linear equalizer are provided. Through comparisons, some asymptotic results for the performance in the perceptron-based equalizer are obtained. Furthermore, a comparison of the performance of the perceptron-based equalizer (including the decision feedback equalizer) with the least mean squares linear transversal equalizer (including decision feedback equalizer) indicates that the former offers significant reduction in the bit error rate. This is because it has the ability to form highly nonlinear decision regions, in contrast with the linear equalizer which only forms linear decision regions. The linear nature of the decision regions limits the performance of the conventional linear equalizer.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Colin F.N.Cowan, for his inspiration and technical guidance, my second supervisor, Dr. Bernard Mulgrew, and Professor Peter M. Grant, for their enthusiasm, Dr. Gavin J.Gibson for many fruitful technical discussions, and the managing Director of the Telecommunication Labs., Dr. Shyue-Ching Lu for his encouragement. I acknowledge the National Science Council and the Telecommunication Labs., of the Republic of China for providing the financial support. Finally thanks to my wife, Jane, for her encouragement and understanding.

To my mother, and my wife, Jane.

CONTENTS

List of Abbreviations

AWGN	Additive White Gaussian Noise
BER	Bit error ratio
DFE	Decision Feedback Equalizer
FIR	Finite Impulse Response
ISI	Intersymbol Interference
LMS	Least Mean Squares
LTE	Linear Transversal Equalizer
LTI	Linear Time-Invariant
MAP	Maximum a-Posteriori Probability
MLP	Multi-layer Perceptron
MLSE	Maximum Likelihood Sequence Estimation
MSE	Mean Square Error
RLS	Recursive Least Squares
SNR	Signal to Noise Ratio
VLSI	Very Large Scale Integration

List of Principal Symbols

Variables and Constants

d	Estimation lag of equalizer
ϕ	Input dimension
$\{e_n\}$	Error Sequence $e_0, e_1, \ldots, e_n, \ldots,$
$\{h_n\}$	Impulse response sequence $h_0, h_1, \ldots, h_n, \ldots,$
c_i	The i-th tap weight of the LTE
c_i^f	The i-th tap weight of the feedforward equalizer in DFE
c_i^b	The i-th tap weight of the feedback filter in DFE
$w_{ij}^{(m)}$	The weight between i-th node and j-th node at m-th layer perceptron-based equalizer
η_n	The n-th additive Gaussian noise variable at n-th sampling time
e	error
$v_j^{(m)}$	The output of j-th node (neuron) at m-th layer

η	Learning gain in Back Propagation learning algorithm
β	Threshold level gain in Back Propagation learning algorithm
α	Momentum parameter in Back Propagation learning algorithm
$\eta_{eff}(\alpha,\beta)$	Effective learning gain
$\eta^{*}_{eff}(\eta,\alpha)$	The effective learning gain corresponds to the minimum value of BER for a η for SNR=20 dB
L_t	Total number of weights or links in Multi-layer Perceptrons
μ	Learning used in LMS algorithm
$\{y_n\}$	Received signal sequence $y_0, y_1, \ldots, y_n, \ldots,$
$\{u_n\}$	Transmitted signal sequence $u_0, u_1, \ldots, u_n, \ldots,$
$\{\hat{u}_n\}$	Estimated signal sequence $\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_n, \ldots,$
$\{\tilde{u}_n\}$	Decision signal sequence $\tilde{u}_0, \tilde{u}_1, \ldots, \tilde{u}_n, \ldots,$
$p(H_i \mid r)$	Probability that r comes from H_i
$p(r \mid H_i)$	Likelihood function of class H_i
$p(H_i)$	A priori probability of class H_i
$p(e \mid b)$	The error probability depends on b
$p(s_i^+)$ or $p(s_i^-)$	Probability density function of s_i^+ or s_i^-
$P(e)$	Bit error probability or bit error rate

ζ	power metric
ρ_i	Hyperplane formed by the i^{th} neuron in the first hidden layer

Operators

z^{-1}	Unit sample delay
Σ	Summation
Π	Product
T	Unit sample duration
E[·]	Statistical expection of [·]
tr[·]	Trace of the matrix [·]

CHAPTER 1

INTRODUCTION

1.1 General

Equalization [1] for digital communications can be simply described as the task of reconstructing or estimating a data sequence from a set of sequential observations which are made at the output of a communication channel. The need for equalizers arises due to the fact that the channel introduces dispersion, causing inter-symbol interfence (ISI), and noise. A simplistic view of this situation would indicate that the required operation is to form a filter which inverts the channel response, i.e. an inherently linear process. However this approach ignores the problems associated with the fact that such an inverse would basically require to be infinite in nature and also takes no account of in-band noise.

Usually, the communication channel is assumed to be linear and the structure of equalizers which includes both linear and decision feedback equalizers is based on the linear filter [1]. The decision feedback equalizer (DFE) uses the feedback decisions to cancel the intersymbol interference terms. This has traditionally been seen as an non-linear equalizer. Basically however, the filter structure is still linear in nature. The linear filter most often used for equalization is the transversal filter [1],[2]. The

decision regions of a linear equalizer are always limited by hyperplanes. The linearity of these decision boundaries limits the performance of the system (for example data rate and bit error rate), especially in poor noise conditions.

As the received signal sets are not always linearly separable in the case of a non-minimum phase channel, it may be impossible to pass a hyperplane between the received signal sets. In this case equalization cannot be implemented by a linear equalizer without introducing some time delay in the estimation of the transmitted signal so as to make the received signal sets linearly separable. However, non-linear filters are able to form decision regions which are considerably more complex and have highly nonlinear boundaries. As a result nonlinear equalization can equalize both minimum phase and nonminimum phase channels without the introduction of any time delay.

Mathematical analysis of nonlinear systems is very difficult in comparison with linear systems. Most of the properties of the nonlinear systems can only be obtained by simulation. There are a number of nonlinear filter structures such as Volterra series [3],[4],[5],[6] and artificial neural networks [7],[8] but until recently these have been too complicated to implement. Recent advances in computer and VLSI technologies now permit signal processing systems to handle such sophisticated and computationally intensive algorithms. As a result nonlinear filters are becoming more common. The nonlinear filter structures are capable of forming highly nonlinear decision boundaries. The complex nonlinear decision boundary gives a high tolerance to input noise in the case of equalization. As a result the nonlinear equalizer exhibits higher resolution performance in comparison to the linear equalizer, especially in poor signal conditions.

Conventional nonlinear filters, like the Volterra series are based on the Taylor series expansion which introduces nonlinearity by an expansion of the input space dimen-

sionality (A.1*) while still using a linear combiner adaptation structure. The computational complexity requires $O(N^k)$ multiplications per data sample for k^{th} order Volterra series, where N is the dimension of the input space. In practical applications, the order (k) is truncated to a few low order terms. There is a huge increase in complexity which results from the inclusion of even minor nonlinearities. The use of truncated Volterra series ($k \leq 3$) for nonlinear channel equalization in radio and satellite communications and echo cancellation in digital subscriber loops have been reported [9],[10],[11],[12].

More recently attention has been focused on a more general form of nonlinear structure which is in fact based on Rosenblatt's original structure, the so called multi-layer perceptron [13], which is a special case of an artificial neural network. The network possesses a capability to learn and adapts itself to changes in the environment and exhibits a high tolerance to input noise. The multi-layer perceptron has a highly nonlinear behaviour in nature because of the nonlinear activation function in the network structure. At present there is little known concerning the theoretical analysis of such networks. This thesis focuses on this structure and its application to the important practical field of channel equalization in digital communications.

The advantages of artificial neural networks over conventional linear structures are summaried below and will be elaborated in the thesis. The non-linear decision boundaries which can be formed by these networks are close to the optimal decision boundary, therefore the neural network gives an improved bit error rate performance, particularly when operating at low signal to noise ratios. It is also capable of separating received signal sets which are not linearly separable, enabling non-minimum phase

* Appendix A

channels to be equalized without the introduction of a time delay. In addition, it seems likely that nonlinear equalizers may offer improved performance when the channel distortion is also nonlinear. Nonlinearities in the transmission system may be introduced by saturation effects in power amplifiers, analogue/digital interface circuits, inaccuracies in signal companding (compressing-expanding) [14] in telephone networks as well as in the transmission medium itself.

1.2 Layout of Thesis

The thesis is divided into two parts: the first containing chapters 2, 3 and 6, is devoted solely to channel equalization, multi-layer perceptrons, Minkowski-ζ back propagation (where ζ is the power metrics of the error cost function) and learning parameter considerations; the second, containing chapters 4 and 5, concerns the perceptron-based equalizer and perceptron-based decision feedback equalizer for the digital communications channel.

Chapter 2 presents the minimum probability of error criterion, optimal linear bound of error probability, properties of the optimal decision boundary, and properties of the optimal linear decision boundary for channel equalization.

Chapter 3 is concerned with the multi-layer perceptron, emphasizing the architecture, the back propagation learning algorithm, and the network topology.

In chapter 4, a new approach for nonlinear channel equalization using the multi-layer perceptron architecture is discussed. The performance of the perceptron-based equalizer and some of its properties is explored. The performance of the perceptron-based equalizer, in terms of the convergence time, the mean square error (MSE) and the bit error rate, is compared to the theoretical error bound and the performance of the

conventional linear equalizer using least mean squares algorithm.

Chapter 5 presents a study of a new architecture for decision feedback equalization using the multi-layer perceptron architecture, emphasizing architecture, modelling, the learning algorithm, intersymbol interference cancellation analysis, and performance. Comparisons of the performance of this perceptron-based DFE with a conventional DFE are provided.

In chapter 6, a new model of back propagation learning algorithm using Minkowski-ζ power metrics, and some of its properties is discussed. In particular, the effect of ζ on the performance and robustness. Furthermore, an effective learning gain that relates the learning gain and the momentum parameter is given. The effect of various learning gains, and momentum parameters on the performance is studied using the equalizer scenario.

Finally chapter 7 summaries the conclusions that have been drawn and provides suggestions for further investigation.

CHAPTER 2

EQUALIZATION

2.1 Introduction

A baseband digital communication system consists of transmitter filter, communications medium, receiver filter, equalizer and decision device as shown in Figure (2.1). When signals are propagated through a transmission channel they are generally distorted in some way by the channel. A linear dispersive channel consists of transmitter filter, communications medium and receiver filter which can be modelled by a finite impulse response (FIR) filter, as shown in Figure (2.2), where $\{ u_n \}$ are independent symbols from the M-ary set $U = \{ \pm 1, \pm 3, \ldots\ldots \pm (M-1) \}$. The impulse response in the z-transform domain can be expressed as

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + \ldots\ldots + h_k z^{-k} \tag{2.1}$$

The sampled signal observed at the output of the channel can be written as

$$y_n = \sum_{i=0}^{k} h_i \, u_{n-i} + \eta_n \tag{2.2}$$

The sample value can be rewritten as

$$y_n = u_n \, h_o + \sum_{i=1}^{k} h_i \, u_{n-i} + \eta_n \tag{2.3}$$

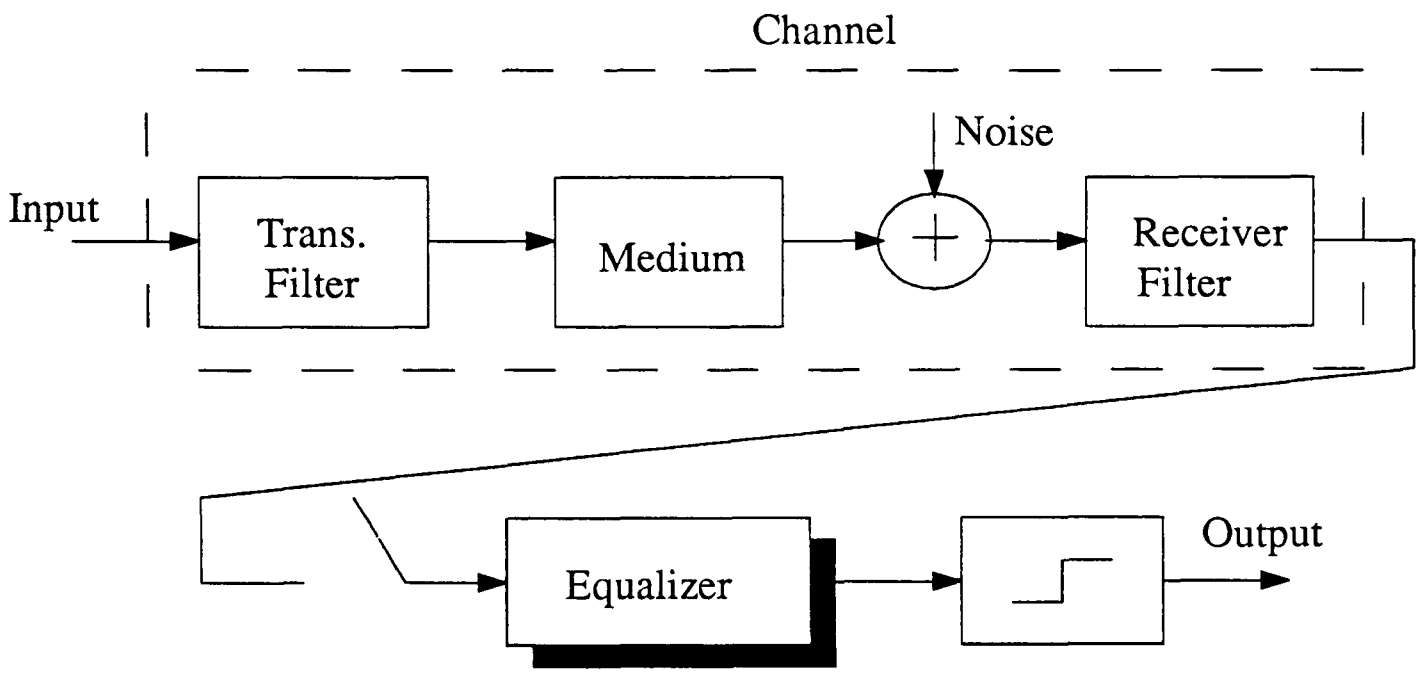


Figure (2.1) A baseband data transmission system

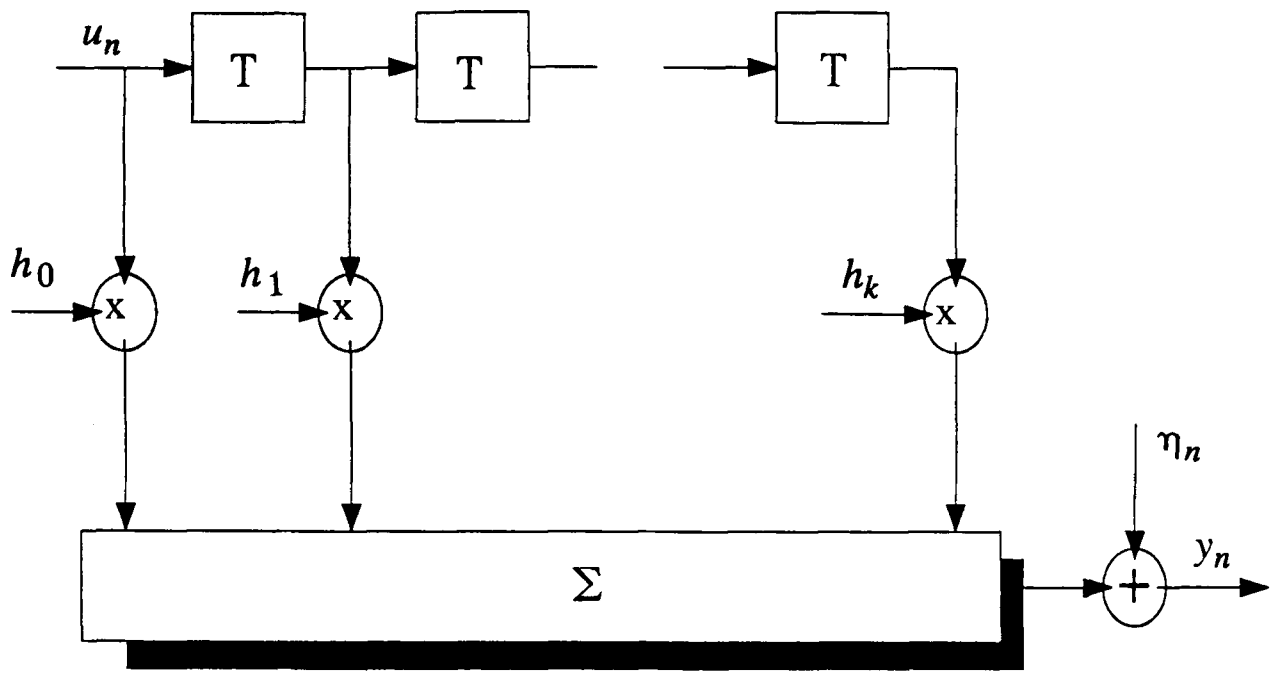


Figure (2.2) Discrete channel model

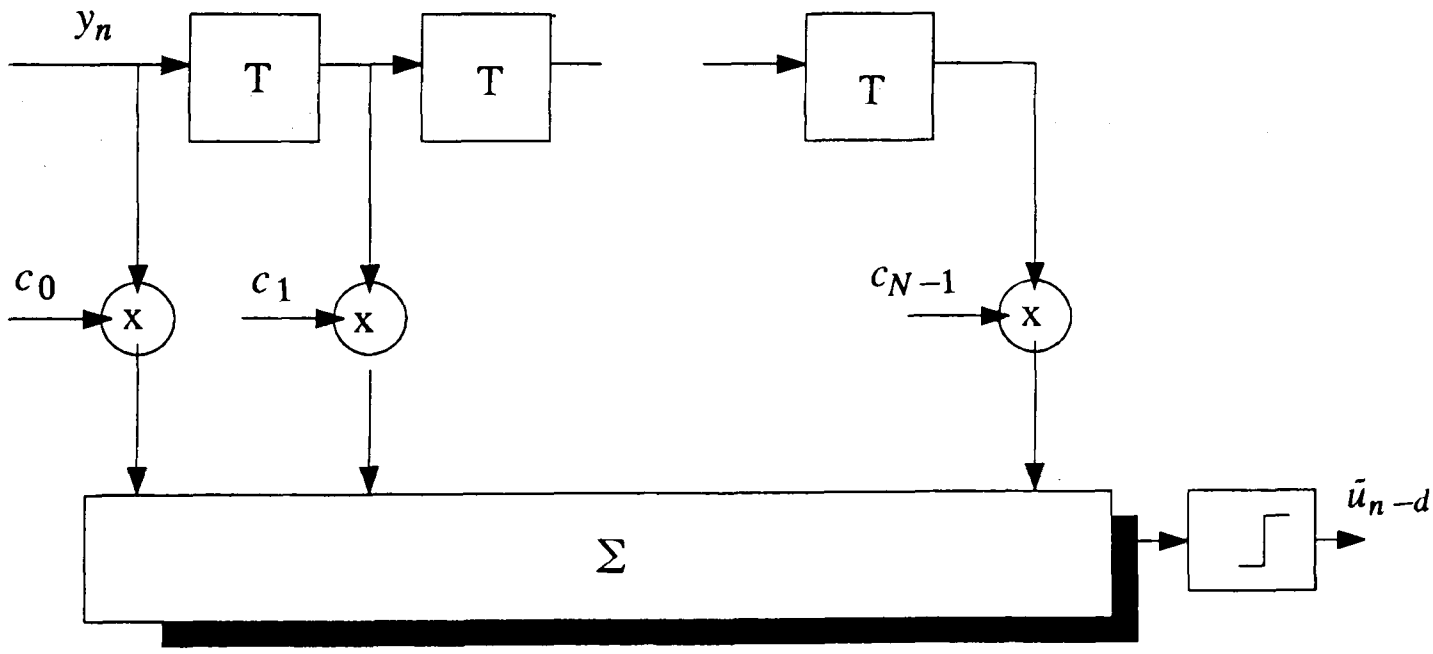


Figure (2.3) Linear transversal equalizer

The term $u_n h_o$ represents the desired signal symbol, the term $\sum_{i=1}^{k} h_i\ u_{n-i}$ represents the intersymbol interference (ISI), and η_n is the additive Gaussian noise sample at the n^{th} sampling instant.

Allowing for a delay of d sample intervals between transmission and reception of a given sample, the above equation can be rewritten as

$$y_n = u_{n-d}\ h_d + \sum_{i \neq n-d}^{k} h_i\ u_{n-i} + \eta_n \tag{2.4}$$

To compensate for the distortion, the received signal $\{ y_n \}$ is fed into a filter whose output will be the estimate $\hat{u}_n$ of the transmitted signal u_n. Such a filter is known as an equalizer.

If the filter is a linear transversal filter i.e., when

$$\hat{u}_{n-d} = \sum_{i=0}^{N-1} c_i\ y_{n-i} \tag{2.5}$$

where the coefficient sequence $\{ c_i \}$ for $0 \leq i \leq N-1$ and N is a finite-order. This is known as linear transversal equalizer (LTE) [16] (Figure (2.3)). From equations (2.3) and (2.5), we can see for an equalizer with order N and a channel response that spans $(k+1)$ symbols, the number of symbols contrubiting to the intersymbol interference is $(N + k - 1)$. If previous decisions $\tilde{u}_n$ are fed back into the equalizer, as in

$$\hat{u}_{n-d} = \sum_{i=0}^{N-1} c_i^f\ y_{n-i} + \sum_{l=1}^{q} c_l^b\ \tilde{u}_{n-d-l} \tag{2.6}$$

We have a decision feedback equalizer [16] (Figure (2.4)), where the feedforward equalizer coefficient sequence is $\{ c_i^f \}$ for $0 \leq i \leq N-1$, the feedback filter coefficient sequence is $\{ c_l^b \}$ for $1 \leq l \leq q$ and both N and q are finite-order.

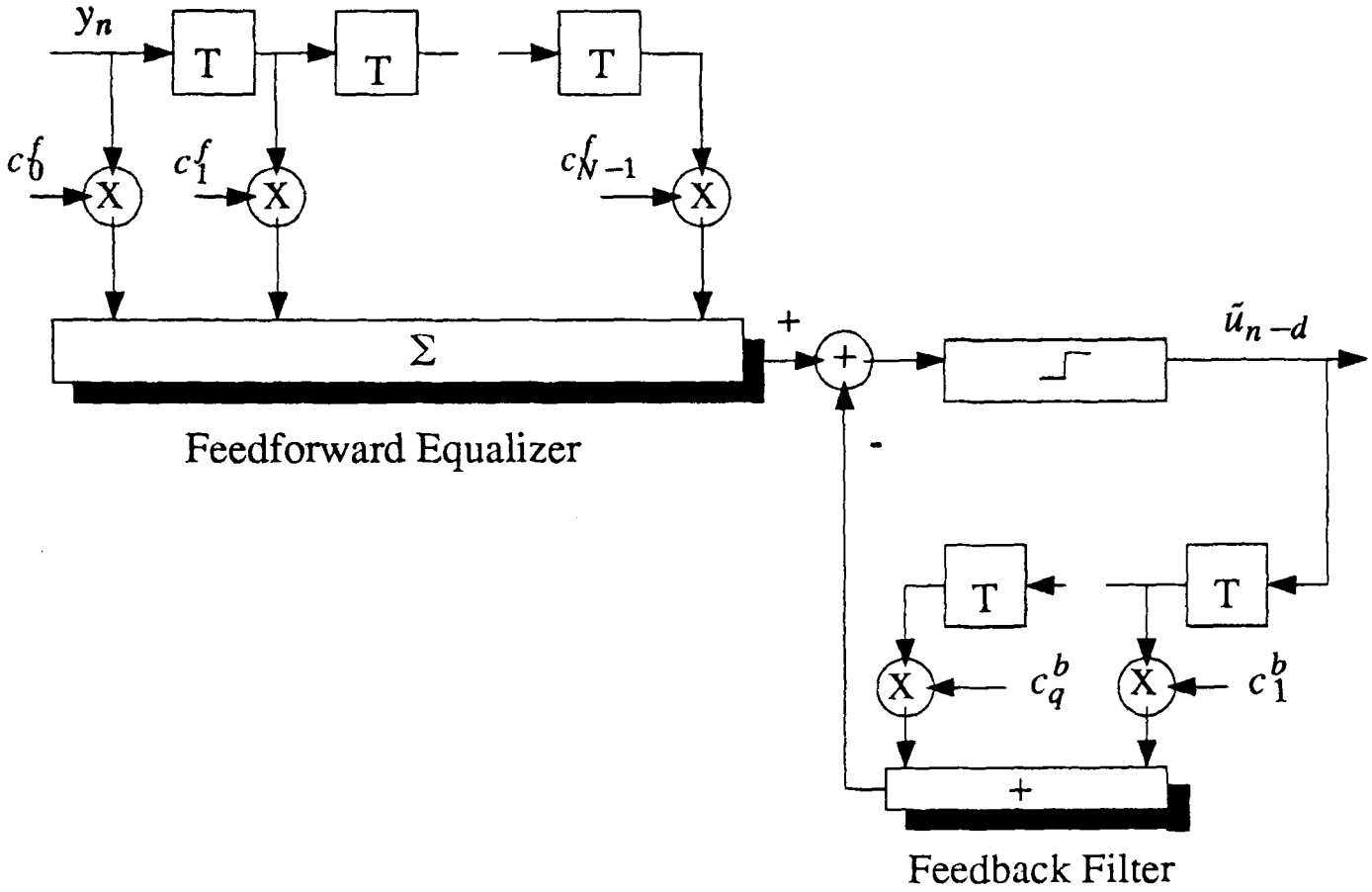


Figure (2.4) Decision feedback equalizer .

The adaptive algorithms often used to adjust the tap vector, $\{ c_0, c_1, \ldots\ldots c_{N-1} \}$, $\{ c_0^f, c_1^f \ldots\ldots\ldots c_{N-1}^f \}$ and $\{ c_1^b, c_2^b \cdots\cdots c_q^b \}$ of the linear transversal equalizer and decision feedback equalizer (DFE) which minimize the output mean square error are the least mean squares (LMS) algorithm [17], and the recursive least squares algorithm (RLS) [17]. In Table (2.1), we present a summary of both the LMS and RLS algorithm. The adaptive algorithms are initialized by use of a training period in which the transmitted symbols are assumed to be known at the receiver. When the residual distortion is small enough, the equalizer is switched into a decision directed mode. In decision directed mode operation, the equalizer can self-adapt to changes in channel characteristics occurring during transmission by using the output of the decision device as a training sequence.

In this chapter, we concentrate on the minimum probability of error criterion, optimal linear bound of error probability, properties of the optimal decision boundary, properties of the optimal linear decision boundary of equalizers, and follow this by examples. As a result of this study, a benchmark is obtained which can be used to evaluate the perceptron-based equalizer performance.

2.2 Optimal Equalizers

This section focuses on the minimum error probability and properties of the decision regions for both the optimal equalizer and the optimal linear equalizer in an additive white Gaussian noise (AWGN) channel. The results are useful in providing a baseline performance for the perceptron-based equalizer. We consider the received signal as a two-dimensional vector (i.e. an observation vector of order 2 is used), because it permits us to represent our results graphically. The results can be extended to higher dimensional vectors.

Table (2.1) Learning Alogrithms

Algorithm	Coefficient Updating	Ancillary Relationships
LMS	$C_n = C_{n-1} + 2\mu V_n e_n$	$e_n = \tilde{u}_n - \hat{u}_n$ $C_n = (c_0, c_1, \ldots, c_{N-1})^T$ $V_n = (y_n, y_{n-1}, \ldots, y_{N-1})^T$ $0 < \mu < \frac{1}{tr[E(V_n V_n^T)]}$
RLS	$C_n = C_{n-1} + K_n e_n$	$K_n = \frac{P_{n-1} V_n}{\lambda + V_n^T P_{n-1} V_n}$ $P_n = \frac{P_{n-1} - K_n V_n^T P_{n-1}}{\lambda}$ $e_n = \tilde{u}_n - \hat{u}_n$ $C_n = (c_o, c_1, \ldots, c_{N-1})^T$ $V_n = (y_n, y_{n-1}, \ldots, y_{N-1})^T$ $\lambda \leq 1$

2.2.1 Minimum probability of error criterion

The received signal on each observation interval may be described as a vector over the interval. The span of the set of possible waveforms on the interval (the set of signal vectors), is known as the signal space. In the signal space, let $S^+ = \{ s_1^+ \ .....s_{n+}^+ \}$ denote the set of signal vectors that may represent the symbol value "1", and let $S^- = \{ s_1^- \ .....s_{n-}^- \}$ denote those may represent "-1". The binary detection problem is stated as

$$H_+ : \qquad r = s^+ + n \tag{2.7}$$

$$s^+ \epsilon S^+ = \left\{ s_1^+ .....s_{n+}^+ \right\} \tag{2.8}$$

and

$$H_- : \qquad r = s^- + n \tag{2.9}$$

$$s^- \epsilon S^- = \left\{ s_1^- .....s_{n-}^- \right\} \tag{2.10}$$

where r is a two-dimensional received signal vector, n represents the noise components with variance σ_v^2.

If $p(H_i|r)$, $i = +, -$ denotes the probability that H_i was the true hypothesis given a particular value of the observation r, we decide that the correct hypothesis is the one corresponding the larger of the two probabilities. The decision rule will then be to choose H_+ if

$$p(H_+|r) > p(H_-|r) \tag{2.11}$$

and choose H_- otherwise. This criterion is known as the maximum a-posteriori probability (MAP) criterion and is identical to the minimum probability of error criterion [18],[19],[20]. Using Baye's rule, the general MAP decision rule using the hypothesis

notation can be written as

$$\frac{p(r|H_+)p(H_+)}{p(r|H_-)p(H_-)} \underset{H_-}{\overset{H_+}{\gtrless}} 1 \tag{2.12}$$

Given an received vector r, the conditional probability density of r is

$$p(r|H_+) = \frac{1}{2\pi\sigma^2} \sum_{i=1}^{n_+} e^{-\frac{|r-s_i^+|^2}{2\sigma_v^2}} p(s^+ = s_i^+ | H_+) \tag{2.13}$$

and

$$p(r|H_-) = \frac{1}{2\pi\sigma_v^2} \sum_{i=1}^{n_-} e^{-\frac{|r-s_i^-|^2}{2\sigma_v^2}} p(s^- = s_i^- | H_-) \tag{2.14}$$

To simplify the mathematics, we assume $p(H_+) = p(H_-)$ and $n_+ = n_- = N'$. The signal vector s_i^+ within the subset S^+ and s_i^- within the subset S^- are equally likely to occur. Since $p(H_+) = p(H_-)$ therefore the MAP decision rule is to choose the received signal as " 1 " when

$$p(r|H_+) > p(r|H_-) \tag{2.15}$$

and " -1 " when

$$p(r|H_-) > p(r|H_+) \tag{2.16}$$

Thus the decision regions which give the minimum probability of error can be defined as

$$H_+ = \left\{ r: \frac{1}{2\pi\sigma_v^2} \sum_{i=1}^{N'} [e^{-\frac{|r-s_i^+|^2}{2\sigma_v^2}} - e^{-\frac{|r-s_i^-|^2}{2\sigma_v^2}}] > 0 \right\} \tag{2.17}$$

and

$$H_- = \left\{ r: \frac{1}{2\pi\sigma_v^2} \sum_{i=1}^{N'} [e^{-\frac{|r-s_i^+|^2}{2\sigma_v^2}} - e^{-\frac{|r-s_i^-|^2}{2\sigma_v^2}}] < 0 \right\} \tag{2.18}$$

The optimal decision boundary line is ξ and is defined by

$$\xi = \left\{ r: \ \frac{1}{2\pi\sigma_v^2} \sum_{i=1}^{N'} [e^{-\frac{|r-s_i^+|^2}{2\sigma_v^2}} - e^{-\frac{|r-s_i^-|^2}{2\sigma_v^2}}] = 0 \right\} \tag{2.19}$$

2.2.2 Optimal linear bound of error probability

For the binary case, we assume the signal sets S^+ and S^- are separable. Then it is possible to pass a hyperplane between the two sets. The regions H_+ and H_- give the minimum probability of error. The hyperplane is the optimal decision boundary for the linear equalizer.

The distance from any signal vector [21] s_i^+ (or s_i^-) to the hyperplane through the points a and b is D_i. Where

$$D_i = \min_{\lambda \in R} d(\lambda) = \min_{\lambda \in R} \left\{ [s_i^+ - \lambda(b-a) - a]^T [s_i^+ - \lambda(b-a) - a] \right\}^{\frac{1}{2}} \tag{2.20}$$

where λ is a constant and real (R).

If we set to zero the derivative of d(λ) with respect to λ , we obtain the fact that

$$D_i = \left\{ \frac{(s_i^+ - a)^T (s_i^+ - a)(b-a)^T (b-a) - [(s_i^+ - a)^T (s_i^+ - a)]^2}{(b-a)^T (b-a)} \right\}^{\frac{1}{2}} \tag{2.21}$$

Where D_i is the perpendicular distance from the signal vector s_i^+ to the hyperplane, and $a \neq b$.

If the signal sets S^+ and S^- are symmetric the hyperplane passes through the origin. The perpendicular distance D_i from s_i^+ to the hyperplane can be simplified as ($a = 0$)

$$D_i = \left\{ \frac{[(s_i^+)^T s_i^+][b^T b] - [s_i^+ b]^T [s_i^+ b]}{b^T b} \right\}^{\frac{1}{2}} \tag{2.22}$$

We can transform the coordinates as shown in Figure (2.5), with the y coordinate parallel to the hyperplane, only the x coordinate would affect the decision. Therefore we can simplify the error probability in terms of the x coordinate only. The total probability of error is therefore:

$$p(e|b) = p(H_+)\sum_{i=1}^{n^+} p(e|s_i^+)p(s_i^+) + p(H_-)\sum_{i=1}^{n^-} p(e|s_i^-)p(s_i^-) \tag{2.23}$$

where

$$p(e|s_i^+) = \frac{1}{(2\pi\sigma_v^2)^{\frac{1}{2}}}\int_{D_i}^{\infty} e^{-\frac{x^2}{2\sigma_v^2}} dx$$

$$= Q\left(\frac{D_i}{\sigma_v}\right) \tag{2.24}$$

Assume that each of the signal vectors are equally likely to occur, and $p(H_+) = P(H_-)$, the total error probability can be rewritten as

$$p(e|b) = \frac{2}{l}\sum_{i=1}^{l/2} Q(D_i/\sigma_v) \tag{2.25}$$

where l is the total number of signal vectors ($S^+ \ U \ S^-$). By using the appropriate value of b, the probability of error can be minimized. To find the vector b that minimizes $p(e|b)$. We need to solve $\frac{\partial p(e|b)}{\partial b} = 0$. Using Leibnitz's rule [22], for differentiating the integral equation (2.25), we obtain

$$\sum_{i=1}^{l/2} e^{-\frac{D_i}{2\sigma_v^2}} \left\{ \frac{(b_1+b_2)^2(s_{i1}^+ b_2 - s_{i2} b_1)s_{i2}^+ + (s_{i1}^+ b_2 - s_{i2}^+)^2 b_1}{D_i} \right\} = 0 \tag{2.26}$$

and

$$\sum_{i=1}^{l/2} e^{-\frac{D_i}{2\sigma_v^2}} \left\{ \frac{(b_1+b_2)^2(s_{i1}^+ b_2 - s_{i2}^+ b_1)s_{i1}^+ - (s_{i1}^+ b_2 - s_{i2}^+ b_1)^2 b_2}{D_i} \right\} = 0 \tag{2.27}$$

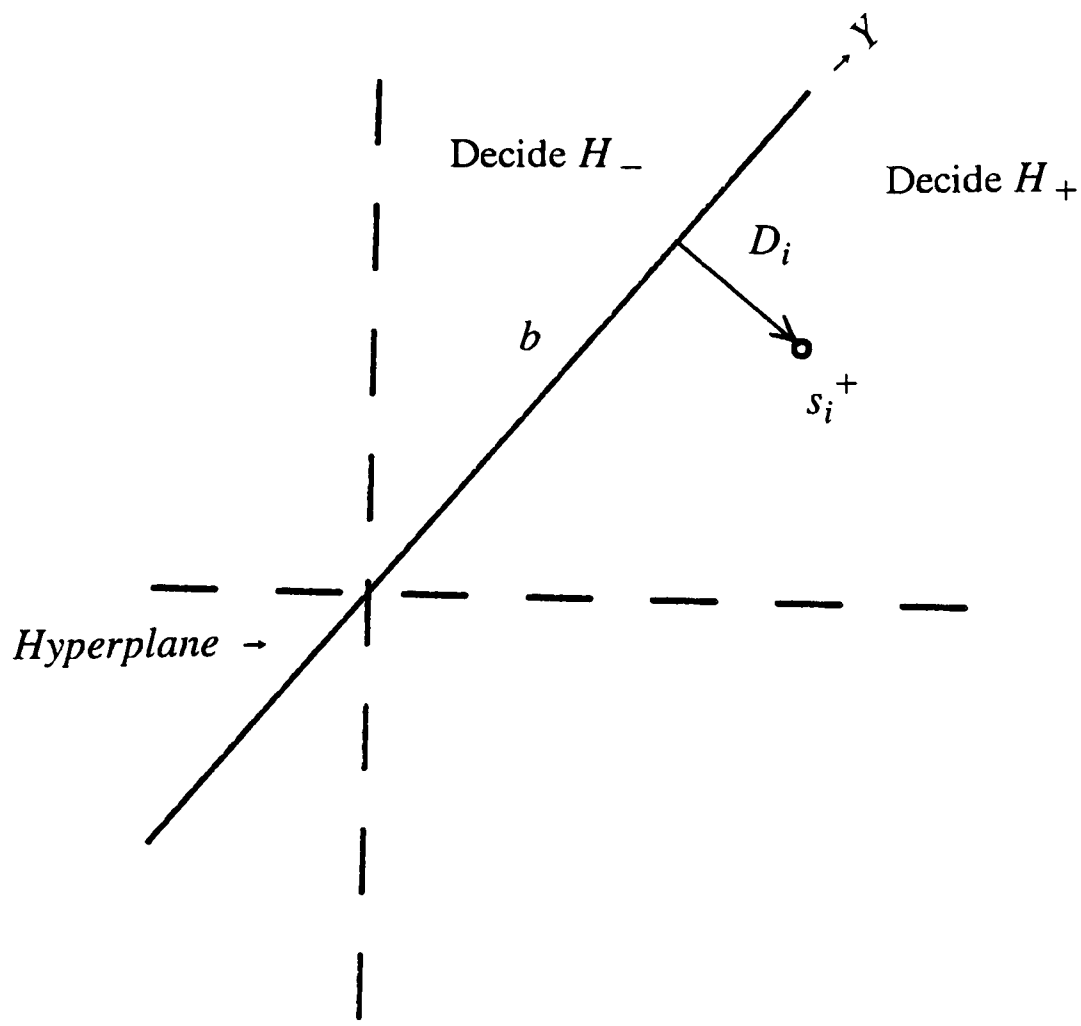


Figure (2.5) Decision space

where

$$b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{2.28}$$

$$s_i^+ = \begin{bmatrix} s_{i1}^+ \\ s_{i2}^+ \end{bmatrix} \tag{2.29}$$

Using a numerical iteration method, the vector b in equations (2.26) and (2.27) can be obtained. Substituting the value of b into equation (2.22), with equation (2.25), the value for the minimum probability of error can be calculated. The above results are not based on any particular equalizer structure, but rather on a general binary hypothesis description of the detection problem. It is also realized that, so far, we have optimized only the decision boundary, not the equalizer system.

2.2.3 Examples

When the roots of the channel transfer function

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + \ldots\ldots + h_k z^{-k} \tag{2.30}$$

all lie strictly within the unit circle in the z-plane, the channel is said to be minimum phase, otherwise it is non-minimum phase. For a three-tap channel, when

$$2h_0 + h_2 > \sqrt{(h_1^2 - 4h_0h_2)} > -2h_0 + h_1 \tag{2.31}$$

or

$$-(2h_0 + h_2) < \sqrt{(h_1^2 - 4h_0h_2)} < 2h_0 - h_1 \tag{2.32}$$

the channel is minimum phase. The non-minimum phase channel can be equalized by linear equalizers if some delay, d , is introduced in the estimation of u_n so that, on the n^{th} iteration, the equalizer estimates the input symbol u_{n-d}.

As an example of the error bound of two-tap equalizers, a three-tap channel is used. The digital message applied to the channel was in random bipolar form { -1,1}. The channel output is corrupted by zero mean white Gaussian noise.

If one-bit delay ($d=1$) is introduced in the estimate of the signal, there are 16 possible two-dimensional signal vectors. The signal vectors are given in terms of channel parameters (h_0, h_1 , and h_2) as

$$S^+ = \begin{bmatrix} h_0+h_1+h_2 & h_0+h_1+h_2 \\ h_0+h_1+h_2 & -h_0+h_1+h_2 \\ h_0-h_1+h_2 & h_0+h_1-h_2 \\ h_0-h_1+h_2 & -h_0+h_1-h_2 \\ h_0+h_1-h_2 & h_0+h_1+h_2 \\ h_0+h_1-h_2 & -h_0+h_1+h_2 \\ h_0-h_1-h_2 & h_0+h_1-h_2 \\ h_0-h_1-h_2 & -h_0+h_1-h_2 \end{bmatrix}^T \tag{2.33}$$

and

$$S^- = \begin{bmatrix} -h_0+h_1+h_2 & h_0-h_1+h_2 \\ -h_0+h_1+h_2 & -h_0-h_1+h_2 \\ -h_0-h_1+h_2 & h_0-h_1+h_2 \\ -h_0-h_1+h_2 & -h_0-h_1-h_2 \\ -h_0+h_1+h_2 & h_0-h_1+h_2 \\ -h_0+h_1+h_2 & -h_0-h_1+h_2 \\ -h_0-h_1-h_2 & h_0-h_1-h_2 \\ -h_0-h_1-h_2 & -h_0-h_1-h_2 \end{bmatrix}^T \tag{2.34}$$

Conversely, if there is no delay ($d=0$), the 16 possible signal vectors are

$$S^+ = \begin{bmatrix} h_0+h_1+h_2 & h_0+h_1+h_2 \\ h_0+h_1-h_2 & h_0+h_1+h_2 \\ h_0-h_1+h_2 & h_0+h_1-h_2 \\ h_0-h_1-h_2 & h_0+h_1-h_2 \\ -h_0+h_1+h_2 & h_0-h_1+h_2 \\ -h_0+h_1-h_2 & h_0-h_1+h_2 \\ -h_0-h_1+h_2 & h_0-h_1-h_2 \\ -h_0-h_1-h_2 & h_0-h_1-h_2 \end{bmatrix}^T \tag{2.35}$$

and

$$S^- = \begin{bmatrix} h_0+h_1+h_2 & -h_0+h_1+h_2 \\ h_0+h_1-h_2 & -h_0+h_1+h_2 \\ h_0-h_1+h_2 & -h_0+h_1-h_2 \\ h_0-h_1-h_2 & -h_0+h_1-h_2 \\ -h_0+h_1+h_2 & -h_0-h_1+h_2 \\ -h_0+h_1-h_2 & -h_0-h_1+h_2 \\ -h_0-h_1+h_2 & -h_0-h_1-h_2 \\ -h_0-h_1-h_2 & -h_0-h_1-h_2 \end{bmatrix}^T \tag{2.36}$$

Each of the signal vectors in the signal space$\{ S^+ \bigcup S^- \}$ are equally likely to occur.

2.2.3.1 Non-minimum Phase Case

A three-tap non-minimum phase channel is used, where $h_0 = 0.3482$, $h_1 = 0.8704$, and $h_2 = 0.3482$. One-bit delay ($d = 1$) is introduced in the estimate of the signal. Figure (2.6) shows the signal space and the optimal decision boundary for different values of signal to noise ratio (SNR). Obviously the signal to noise ratio influences the optimal decision boundary. We observe that for high signal to noise ratio the decision boundary has a piecewise linear appearance, becoming progressively smoother as the signal to noise ratio decreases. Furthermore we remark that the optimal decision boundary is certainly nonlinear.

The decision boundary of an linear equalizer of order N is a hyperplane and is $(N-1)$-dimensional. Using a numerical iteration method, the vector $b = [1, 0.027]^T$ for SNR = 3 dB and $b = [1, 0.214]^T$ for SNR = 10 dB were obtained. The optimal linear decision boundary is shown in Figure (2.7).

Figure (2.8) shows the relative position of the optimal linear decision boundary (hyperplane) and the optimal decision boundary in the signal space at SNR = 10 dB. The minimum distance from any signal vector (s_i^+ , and s_i^-) in the signal space ($S^+ \bigcup S^-$) to the optimal decision boundary is greater than the minimum distance

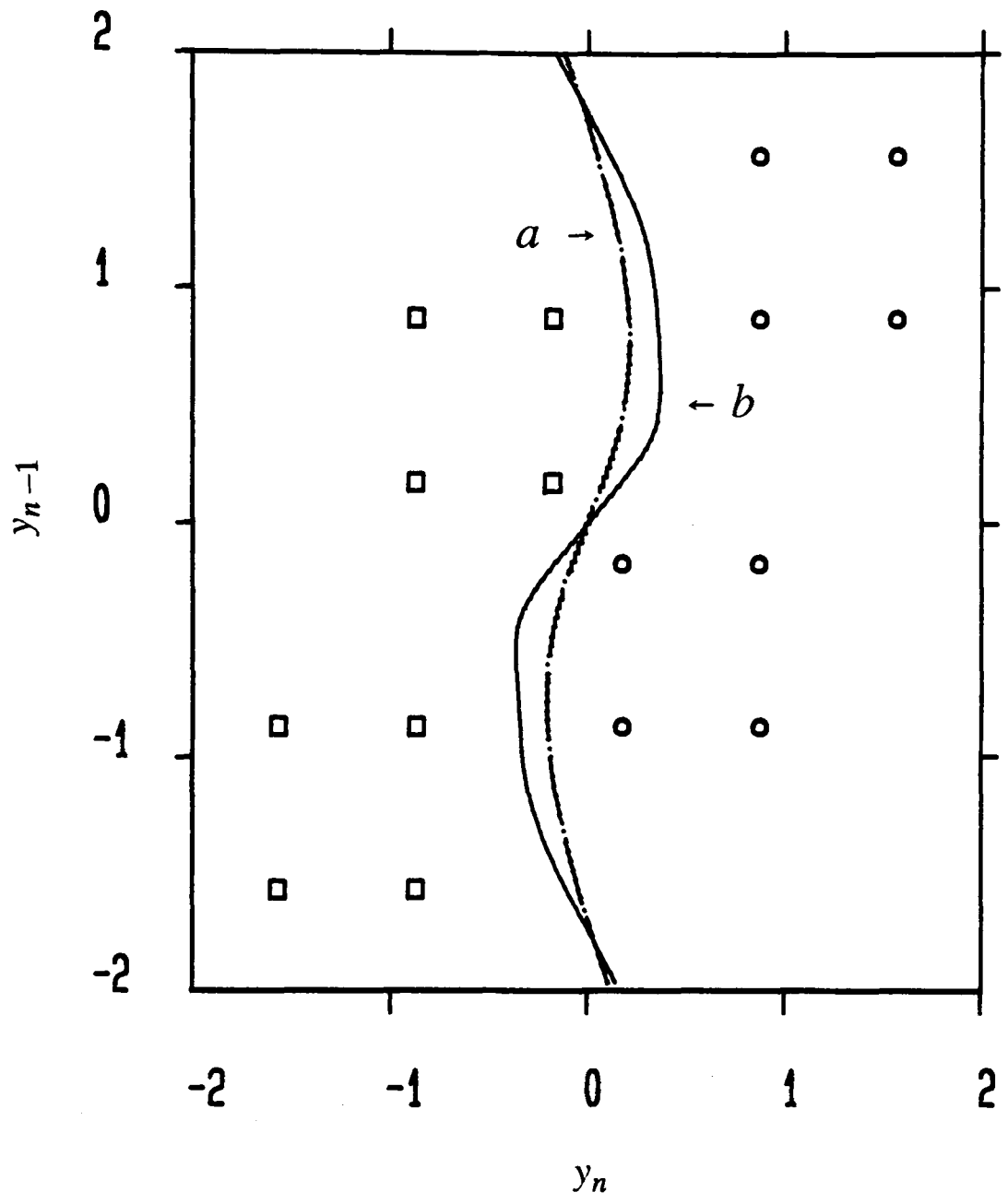


Figure (2.6) Signal space and optimal decision boundary for a non-minimum phase channel with one bit delay ($d = 1$):

(a) SNR = 3 dB,

(b) SNR = 10 dB.

o : signal vector belonging to " 1 ",

□ : signal vector belonging to " -1 ".

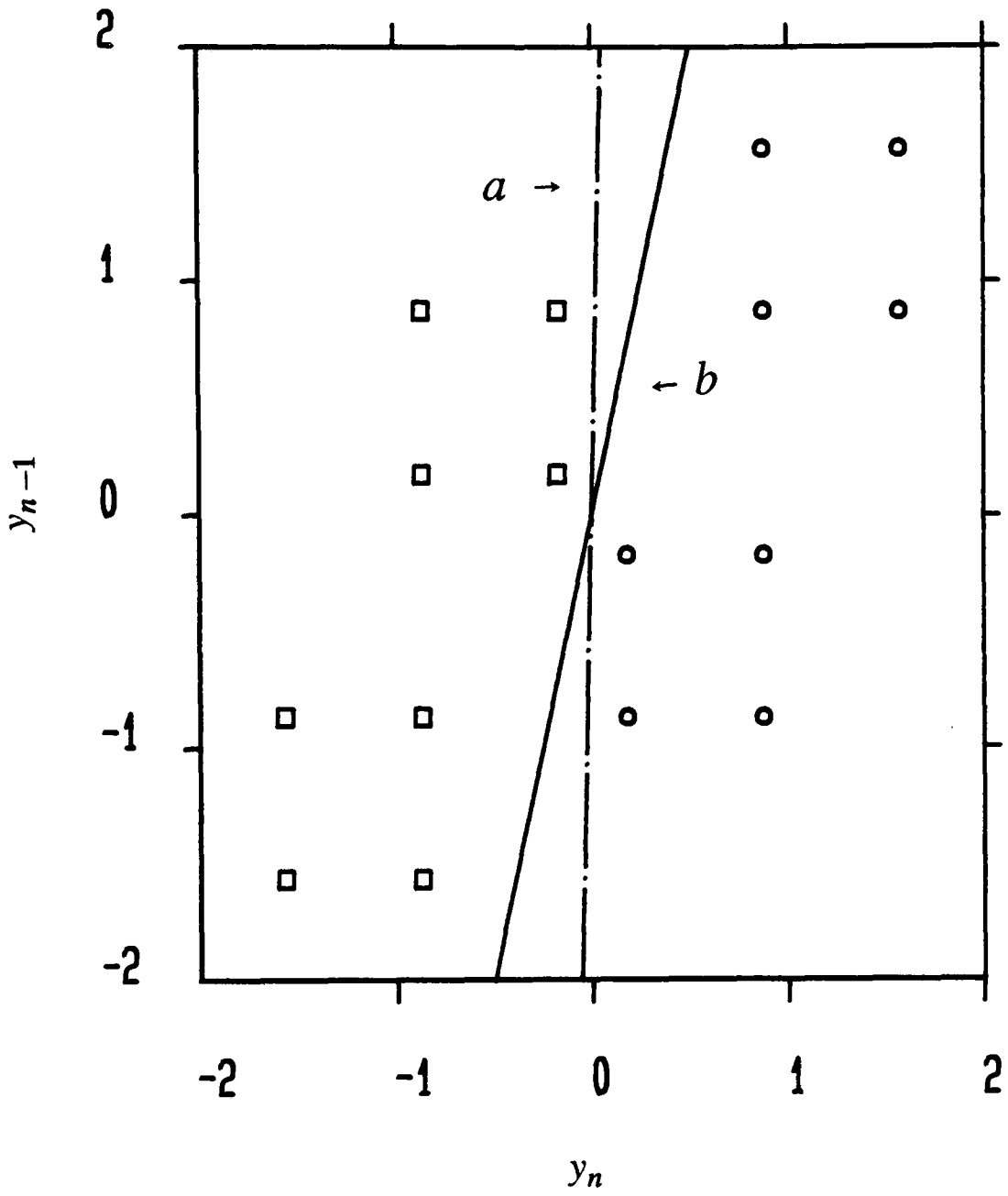


Figure (2.7) Signal space and optimal linear decision boundary for a nonminimum phase channel with one bit delay ($d = 1$):

(a) SNR = 3 dB,

(b) SNR = 10 dB.

o : signal vector belonging to " 1 ",

□ : signal vector belonging to " -1 ".

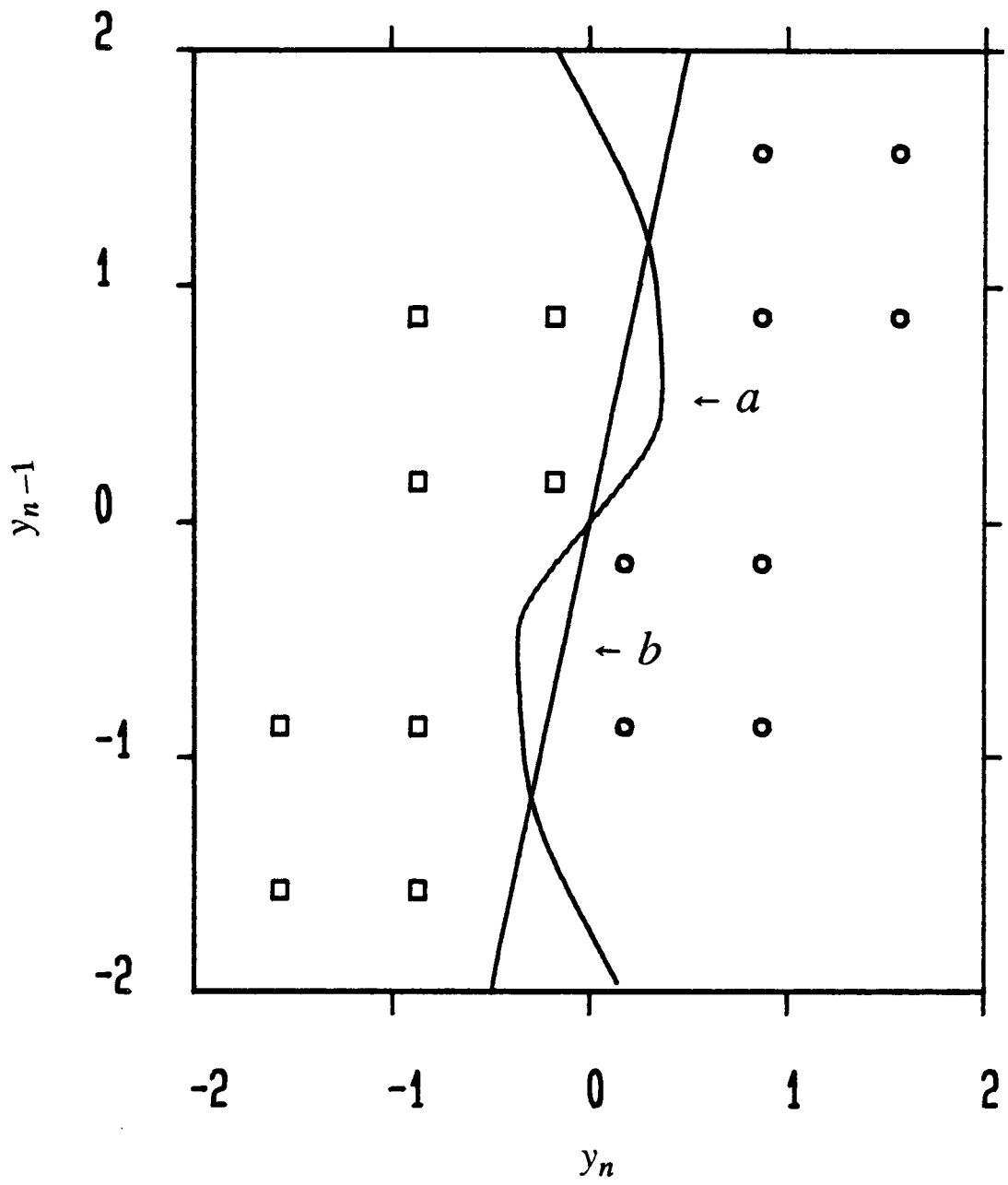


Figure (2.8) The relative position of the decision boundary for SNR = 10 dB

(a) Optimal decision boundary,

(b) Optimal linear decision boundary,

o : signal vector belonging to " 1 ",

□ : signal vector belonging to " -1 ".

from the same signal vector to the hyperplane. The comparison shows graphically why the nonlinear decision boundary provides better performance. The error bound of the optimal equalizer based on equations (2.17) and (2.18), can be obtained by simulation (so called optimal test). The error bound of the optimal linear equalizer based on equation (2.25), can be obtained by computation. Both results are illustrated in Figure (2.9). Examination of Figure (2.9), shows that the optimal equalizer provides at least an improvement of 1.2 dB (SNR) over the optimal linear equalizer.

2.2.3.2 Minimum Phase Case

A two-tap minimum phase channel was used, where $h_0 = 0.894427191$, and $h_1 = 0.447213595$. The signal space and the optimal decision boundary are shown in Figure (2.10) without delay ($d=0$) in the estimate of the signal. Examination of Figure (2.10), shows that for low levels of addition noise the decision boundary has a piecewise linear appearance, becoming progressively smoother as the noise power increases. The optimal decision boundary is certainly nonlinear. Figure (2.11) compares the error bound achieved by the optimal equalizer for both $d=0$ and $d=1$ by simulation. The result indicated that the introduction of one bit delay slightly degraded the performance.

2.3 Discussion

Assuming perfect knowledge of the channel characteristics, the properties of the optimal decision boundaries and the theoretical values of the bit error rate for both nonlinear and linear equalizers can be calculated. As a result of this computation, a benchmark is obtained which can be used to evaluated the performance of the perceptron-based equalizer. Through comparisons, some asymptotic results for perfor-

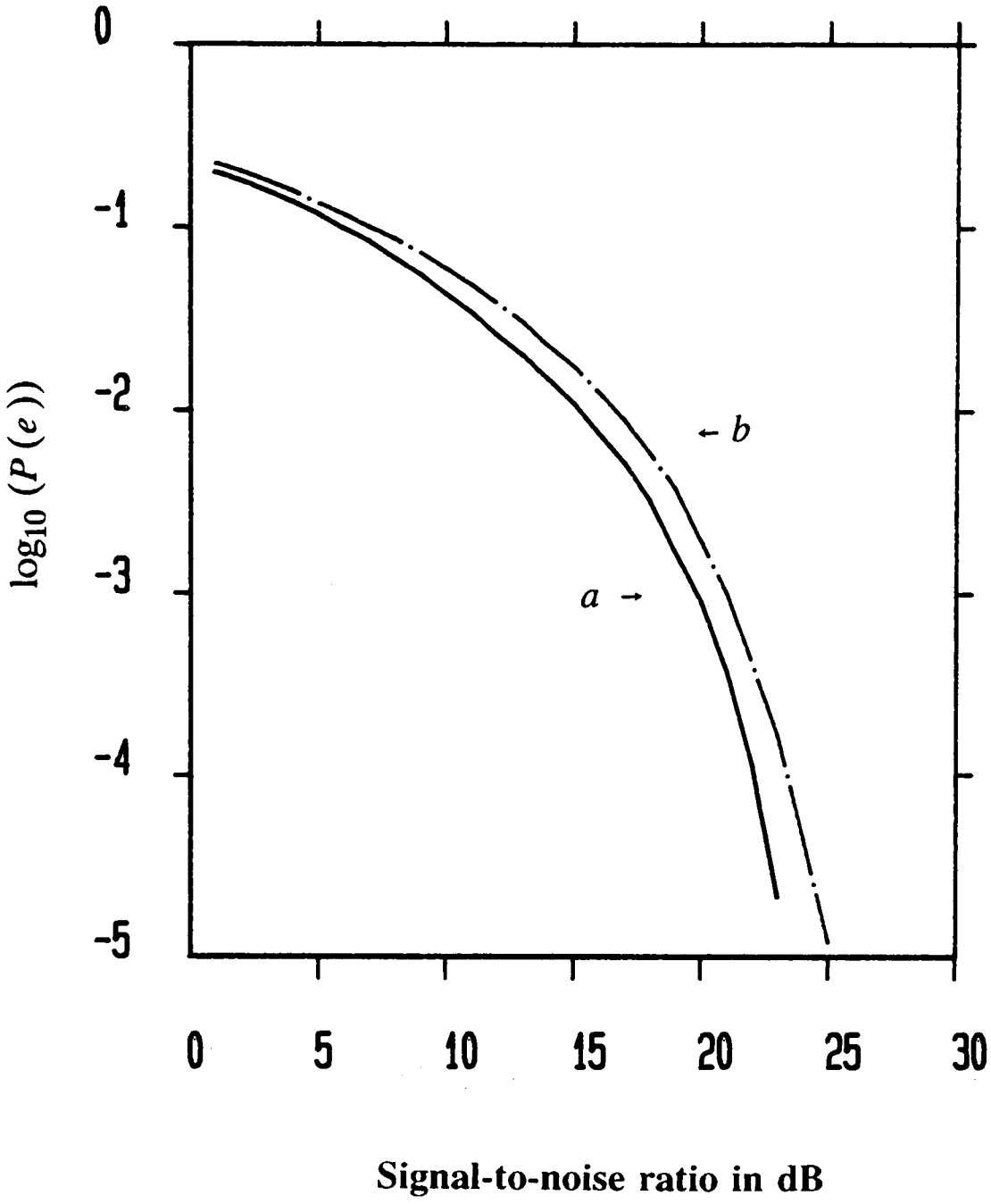


Figure (2.9) The bit error rate performance
(a) Optimal equalizer by simulation,
(b) Optimal linear equalizer by computation.

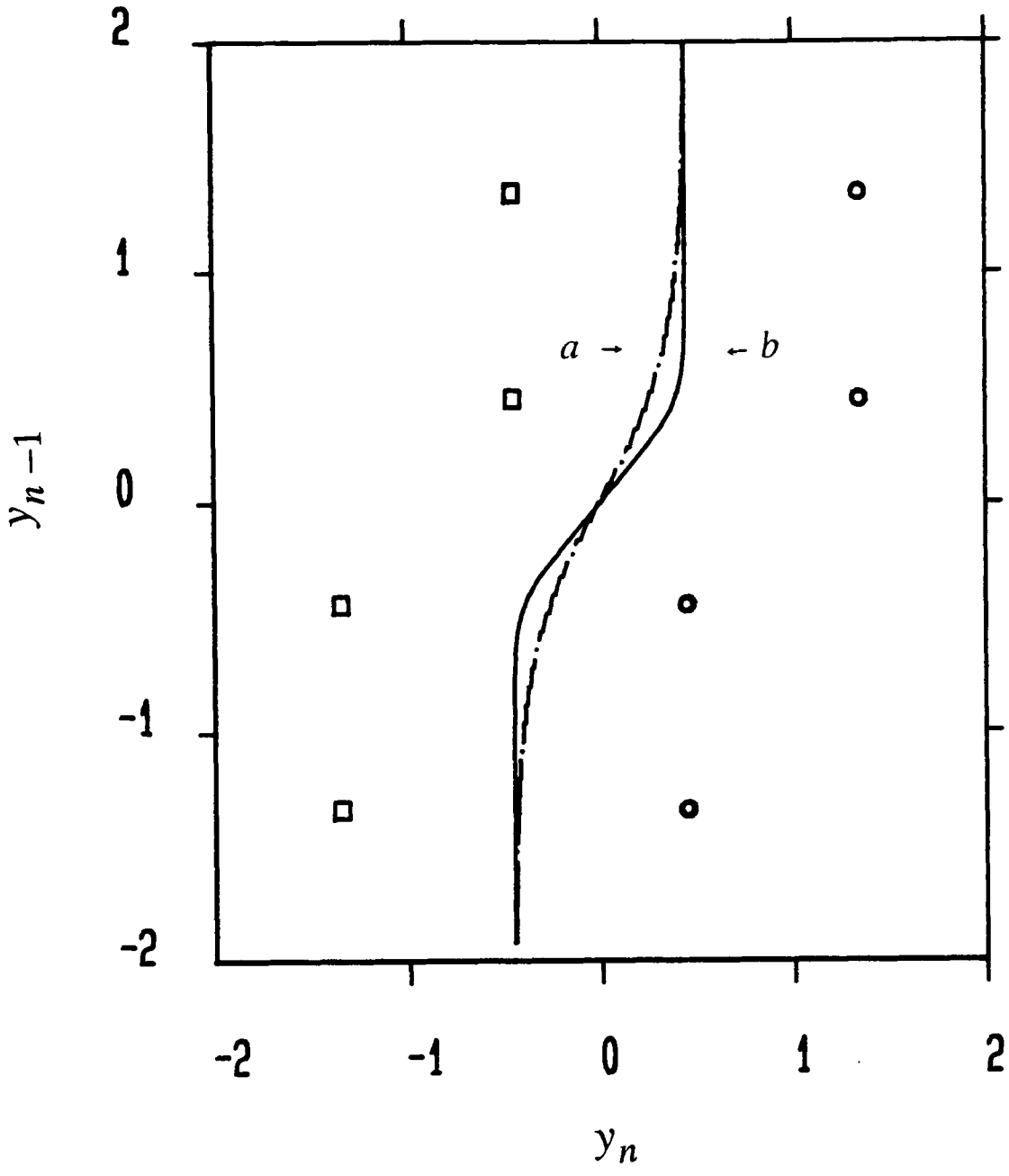


Figure (2.10) The optimal decision boundary for a minimum phase channel without delay ($d = 0$):

(a) SNR = 3 dB,

(b) SNR = 10 dB.

o : signal vector belonging to " 1 ",

□ : signal vector belonging to " -1 ".

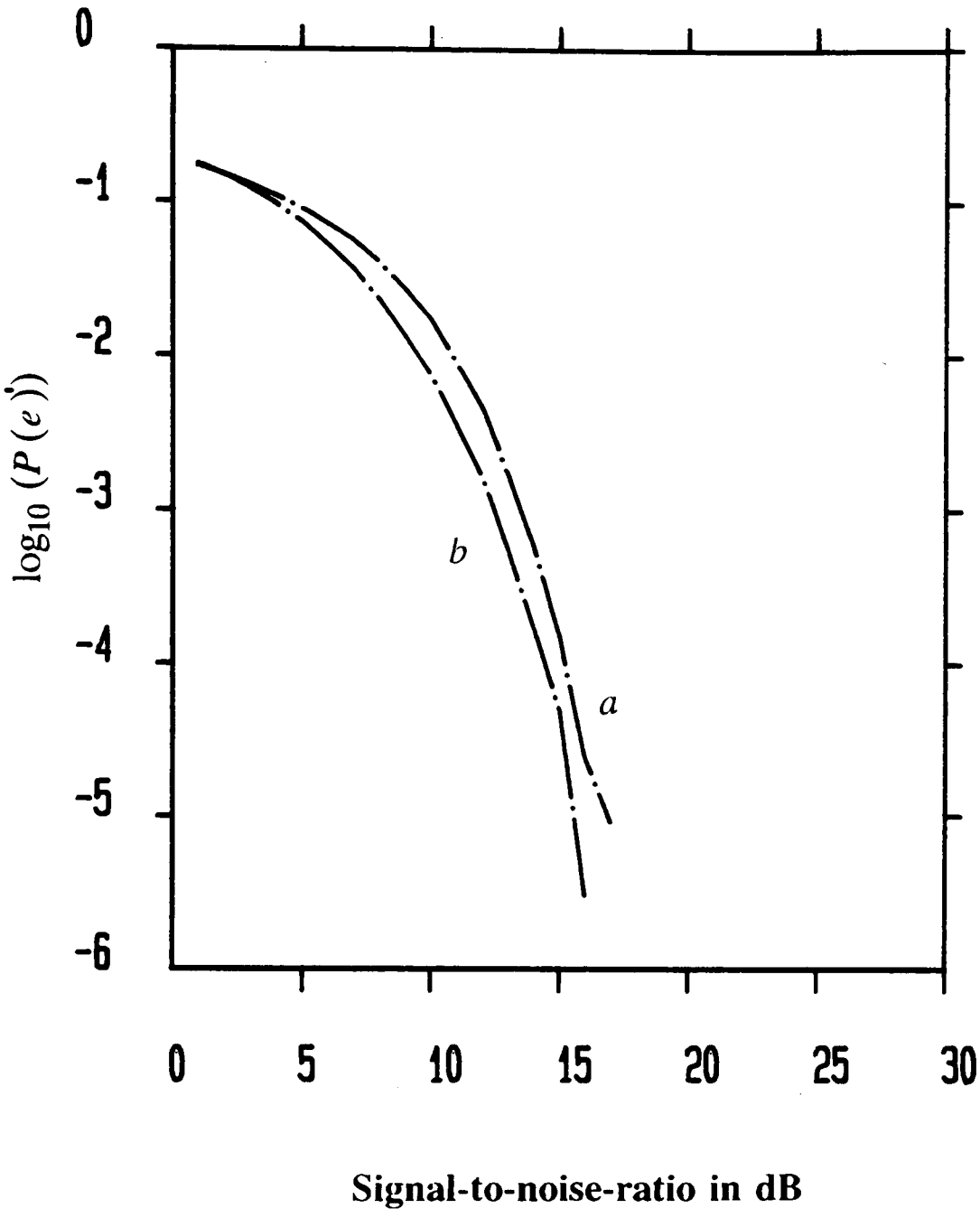


Figure (2.11) Simulation results showing the relative bit error rate performance for a minimum phase channel:

(a) Optimal second order equalizer with one bit delay ($d=1$),

(b) Optimal second order equalizer without delay ($d=0$).

mance in the perceptron-based equalizer can be obtained.

CHAPTER 3

MULTI-LAYER PERCEPTRONS

AND

THE BACK PROPAGATION MODEL

3.1 Introduction

Neural networks are systems which use nonlinear computational elements to model neural behaviour based on our present understanding of the biological nervous system. Knowledge in these models is distributed over many processing elements, so called neurons, and the behaviour of the network in response to the input is a collective decision based on the exchange of information amongst the neurons.

Interest in neural networks began in 1940's with the work of McCulloch and Pitts [23], Hebb [24], Rosenblatt [25], Widrow and Hoff [26], and Minsky and Papert [27]. More recent works include those of Hopfield [28,29,30], Rumelhart and McCelland [31], Sejnowski [32], Feldman [33], and Grossberg [34,35].

In recent years, new learning algorithms [31] for neural networks have appeared and the advances in computer and VLSI technologies have encouraged a broad range of

applications [36-44] including image processing, machine vision, speech recognition, pattern recognition, classification, communications systems (equalization and echo cancelling), biomedical systems and robot control.

A single-layer perceptron [26], or adaptive linear neuron (Adaline) is depicted in Figure (3.1). The output u_n can be expressed as the convolution of the input sequence $\{ x_n \}$ with the weight coefficient sequence $\{ w_n \}$, where $\dot{u}_n = \sum_{i=0}^{N-1} w_i \, x_{n-i} + \dot{q}$, and $\dot{q}$ is the bias. The output can be binary ($\pm$ 1) or analogue. The LMS learning algorithm is used to adjust the weight coefficient sequence. The decision regions of the Adaline is always limited by hyperplanes. The linearity of these decision boundaries limits the performance and also the application of the network. Primary applications are in the field of telecommunications, for example adaptive equalization, echo cancellation and pattern recognition.

A Hopfield network [7],[45] is shown in Figure (3.2). The network has feedback paths from the output back to the input, a so called recursive network. For a given input, the output is calculated and fed back to modify the input. The output is then recalculated and the process is repeated until convergence. The input at $t=0$ is $\dot{u}_i(0) = x_i$, where $0 \leq i \leq N-1$, N is the dimension of the input. The output of the i^{th} node can be expressed as $\dot{u}_j(t+1) = f\left(\sum_{j=0}^{N-1} t_{ij} \, \dot{u}_i(t)\right)$, where $0 \leq j \leq N-1$. The function $f(\cdot)$ is a hard limiting nonlinearity. The weight coefficient sequence $\{ t_{ij} \}$ must be set in advance [7] and does not change during the iteration process (does not learn). The primary applications of the network is to retrieval of complete data or images from fragments. For pattern recognition, the node outputs represent the exemplar pattern that best matches the unknown input.

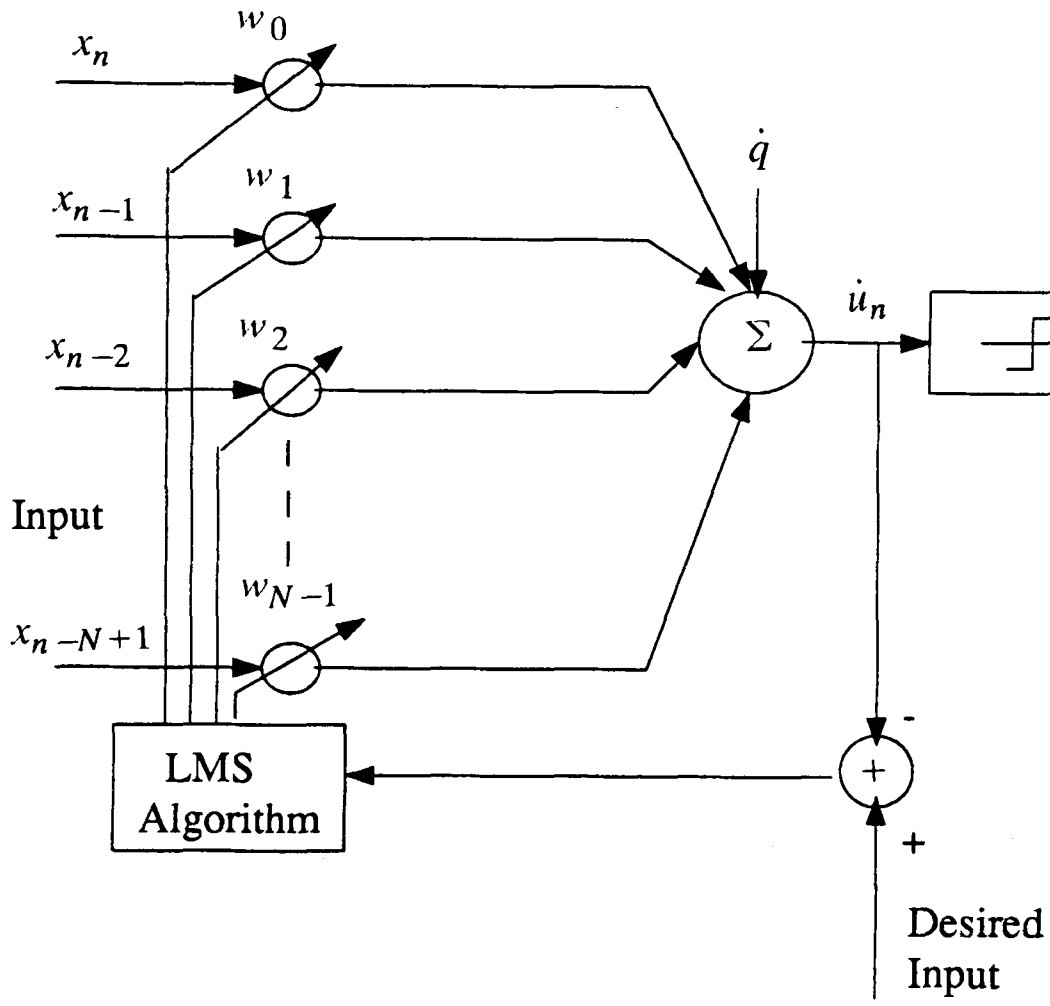


Figure (3.1) Architecture of Single-layer perceptron.

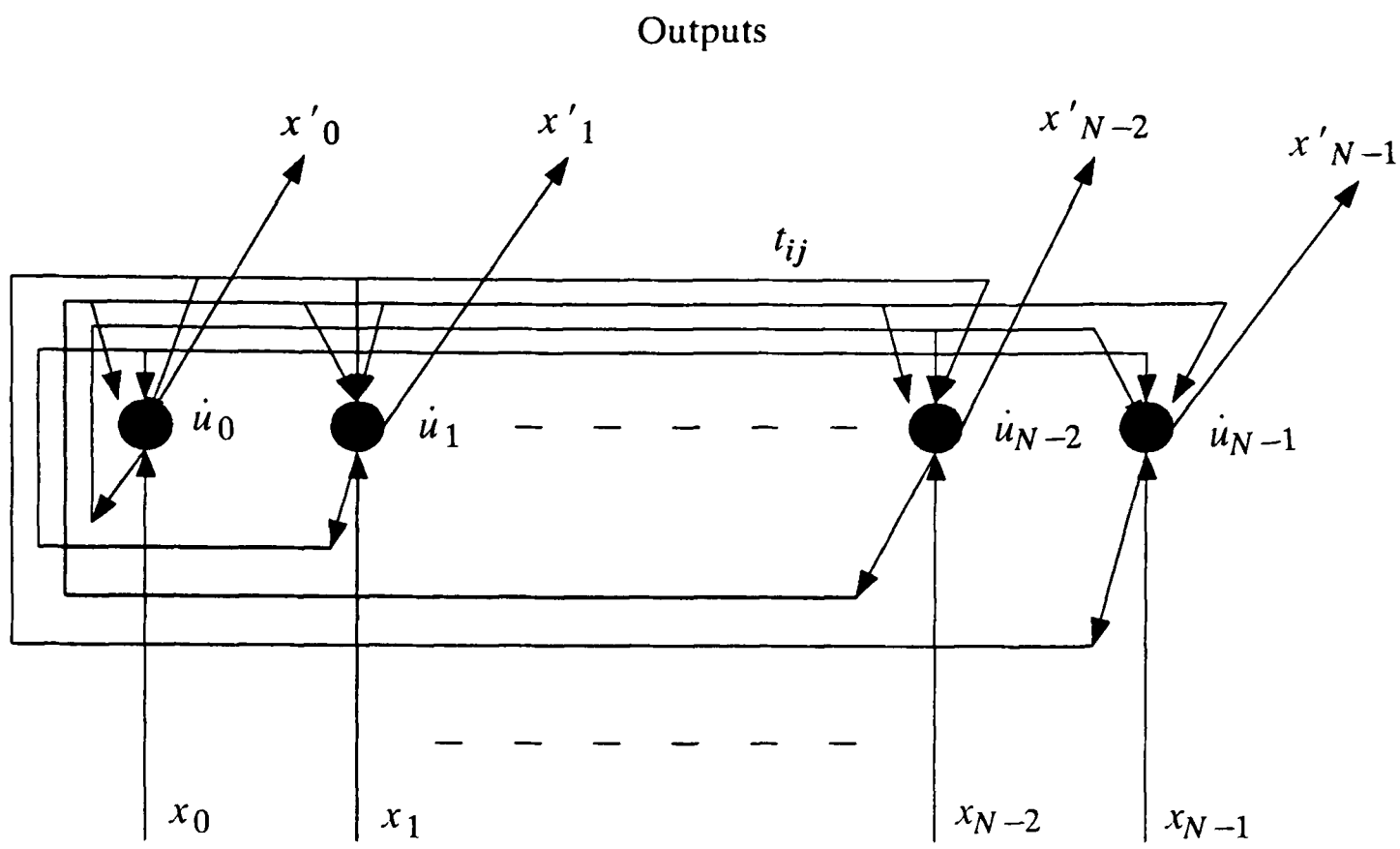


Figure (3.2) Architecture of Hopfield network. $\{ x'_i \}$ are values of output nodes after convergence.

The architecture of the Hamming network [7], [45] is shown in Figure (3.3). The network consists of two subnets; the lower subnet is simply a feedforward network, and the upper subnet is a recursive network. The output of the j^{th} node in the lower subnet is $\dot{u}_j(0) = f(\sum_{i=0}^{N-1} w_{ij}x_i - \theta_j)$, where $0 \leq j \leq M-1$, N is the dimension of the input and M is the dimension of the output. The outputs of the lower subnet will be the inputs of the upper subnet. The output of the j^{th} node is $\dot{u}_j(t+1) = f(\dot{u}_j(t) - \epsilon \sum_{k \neq j} \dot{u}_k(t))$, where $0 \leq j, k \leq M-1$. The process is repeated until convergence. The function $f(\cdot)$ is the threshold logic nonlinearity [7]. The weight coefficient sequence (w_{ij} and ϵ) and thresholds (θ_j) of the network must be set in advance [7], and do not change during iteration. The primary application of the network is for pattern recognition, where N is the dimension of the input pattern, and M is the number of classes.

The Carpenter/Grossberg network [7],[45],[35] is based on adaptive resonance theory [35] and is trained without supervision. The structure of the network is very similar to the Hamming network, except that there are feedback paths from the output nodes to the input nodes to verify the outputs, as shown in Figure (3.4). The input $\{ x_i \}$ is binary ($\pm$ 1) and the network is very sensitive to noise and distortion [7]. The j^{th} output in the lower subnet is $\dot{u}_j(0) = \sum_{i=1}^{N} b_{ij} x_i$, where $j = 1,2,....,M$ (M and N are the dimension of the output and input respectively). The weights $\{ b_{ij} \}$in the lower subnet can be adapted [45]. In the upper subnet there are M nodes representing the dimension of the output. The output in the j^{th} node is $\dot{u}_j(t+1) = f(\dot{u}_j(t) - \epsilon \sum_{j \neq k} \dot{u}_k(t))$, where $j, k = 1,2,3,....,M$, and the function $f(\cdot)$ is threshold logic nonlinearity [7]. The process is repeated until convergence. The weights in the upper subnet ϵ must be set in advance [45]. The primary application is

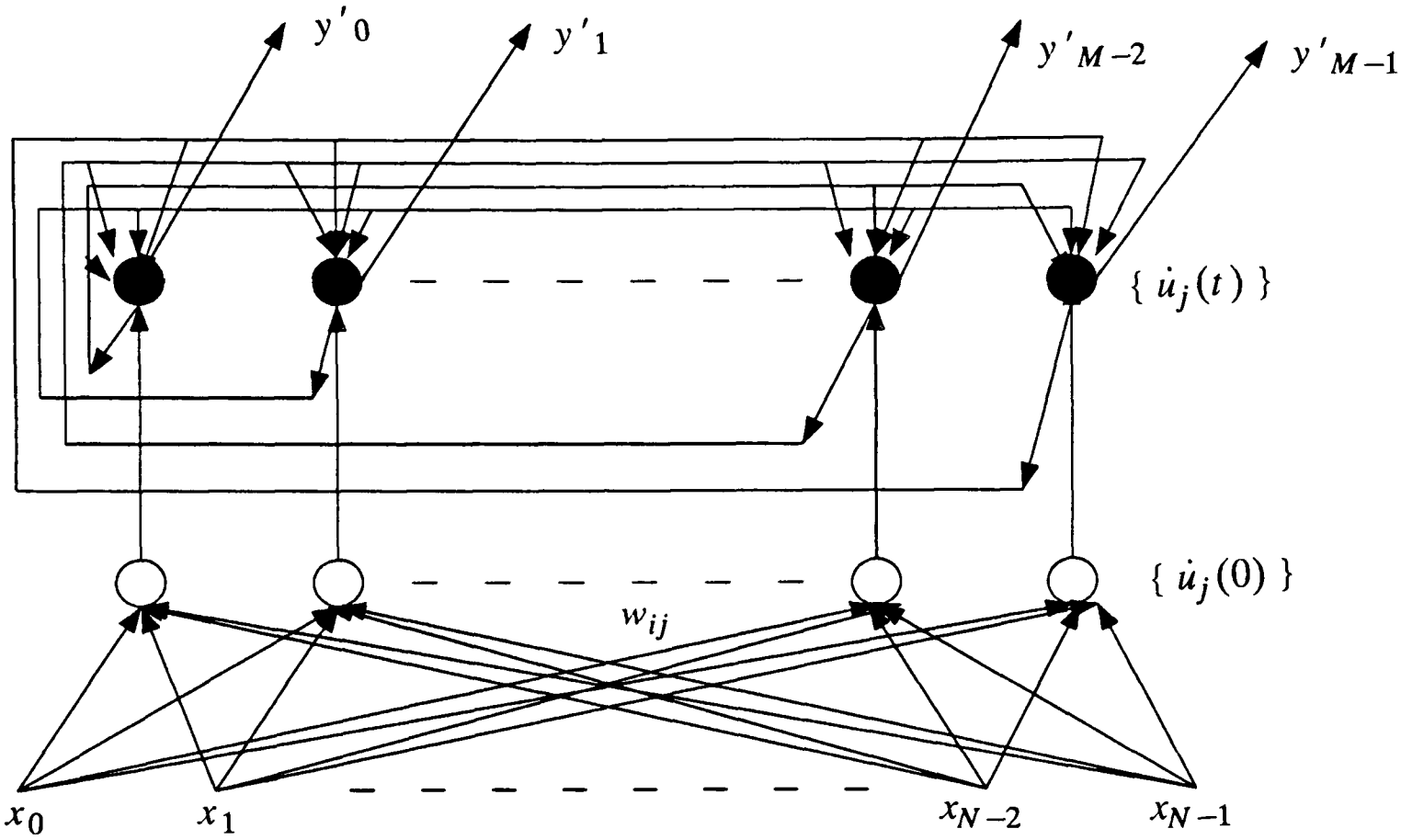


Figure (3.3) Architecture of Hamming network. $\{ y'_j \}$ are values of output nodes after convergence.

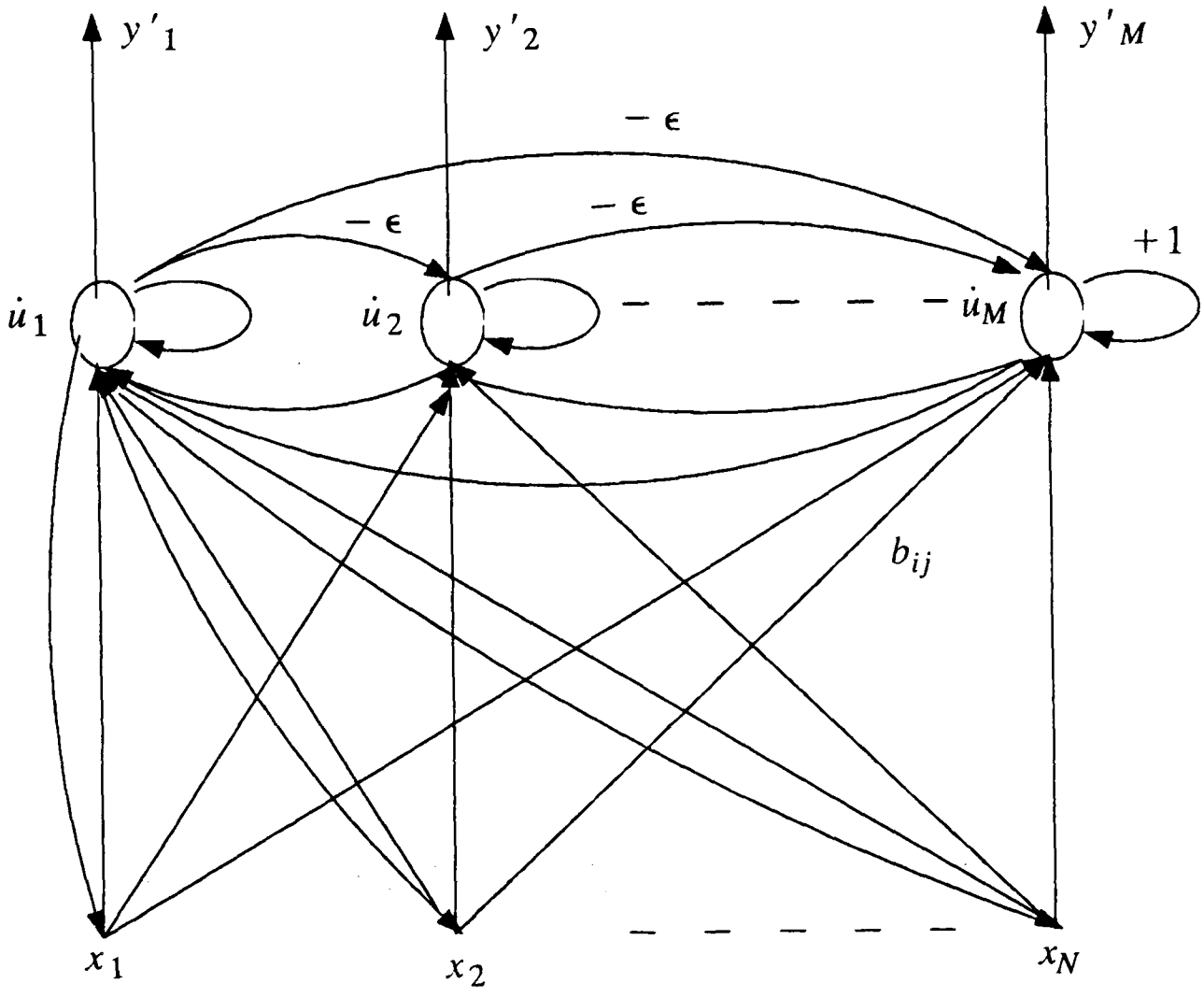


Figure (3.4) Architecture of Carpenter/Grossberg network. $\{ y'_j \}$ are values of output nodes after convergence.

for pattern recognition, where N is the dimension of the input pattern, and M is the number of classes.

Earlier work on single layer perceptrons was limited owing to the fact that only linear decision boundaries (hyperplanes) could be formed in the signal space [25], [26], [27]. Rosenblatt [25] apparently knew the importance of the multi-layer perceptron since he described a wide variety of the architectures studied today. However, at that time the learning algorithm for training the multi-layer perceptron was not available. In 1986, Rumelhart, Hinton, and Williams published their account of how multi-layer perceptrons might be trained with the use of the generalized delta rule (back propagation learning algorithm) [31]. Later developments [31], [45], [46] showed that multi-layer perceptrons are capable of forming more sophisticated decision boundaries (highly nonlinear) than a hyperplane. Because of that, the multi-layer perceptron exhibits a high tolerance to input noise, i.e. it may be applied in poor signal-to-noise ratio conditions.

In single-layer and multi-layer perceptrons, the error information is fedback from the output to modify the weights adaptively; because of that, the learning procedure is robust and self-consistent. Since inputs and adaptive weight coefficients can change over time, the networks adapt and learn. However, in Hopfield and Hamming networks, the weights must be set in advance (fixed weights). Since there are no information feedback to adjust the weights, the networks cannot adapt and learn [7]. Furthermore, the Hopfield, Hamming, and Carpenter/Grossberg networks are used with binary input, and the latter is sensitive to noise.

Throughout the chapter we will only consider multi-layer perceptrons, emphasizing the architecture, the back propagation learning algorithm, and the network topology.

Furthermore, based on the geometric hyperplane analysis, insight is provided into the properties and the complexity of the multi-layer perceptron.

3.2 Multi-layer Perceptrons: Architecture

The neural network's original model is in biological nervous systems, where the basic element is the neuron, which is depicted in Figure (3.5). Each neuron has primarily local connections and is characterized by a set of real weights [w_{1j} .....w_{Nj}] applied to the previous layer to which it is connected and a real threshold level I_j. The j^{th} neuron in the m^{th} layer accepts inputs $V^{(m-1)} \epsilon R^N$ from the $(m-1)^{th}$ layer and returns a scalar $v_j^{(m)} \epsilon R$, given by

$$v_j^{(m)} = f_j\left(\sum_{i=1}^{N} w_{ij}^{(m)}\, v_i^{(m-1)} + I_j^{(m)}\right) \tag{3.1}$$

where $v_i^{(m)} \epsilon V^{(m)}$. The output value $v_j^{(m)}$ serves as input to the $(m+1)^{th}$ layer of the network.

The nonlinearities commonly used in the perceptron are of the sigmoid type, such as

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.2}$$

where $f(x) \epsilon [\ 0,1\ ]$ and

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{3.3}$$

where $f(x)$ lies in the interval [-1,1], the graphs of which are depicted in Figure (3.6). The sigmoid function, $f(x)$ has a real non-zero derivative which makes it useful for stochastic gradient learning methods. The neurons store knowledge or information in the weights $\{\ w_{ij}\ \}$ and the weights are modified through experience or training.

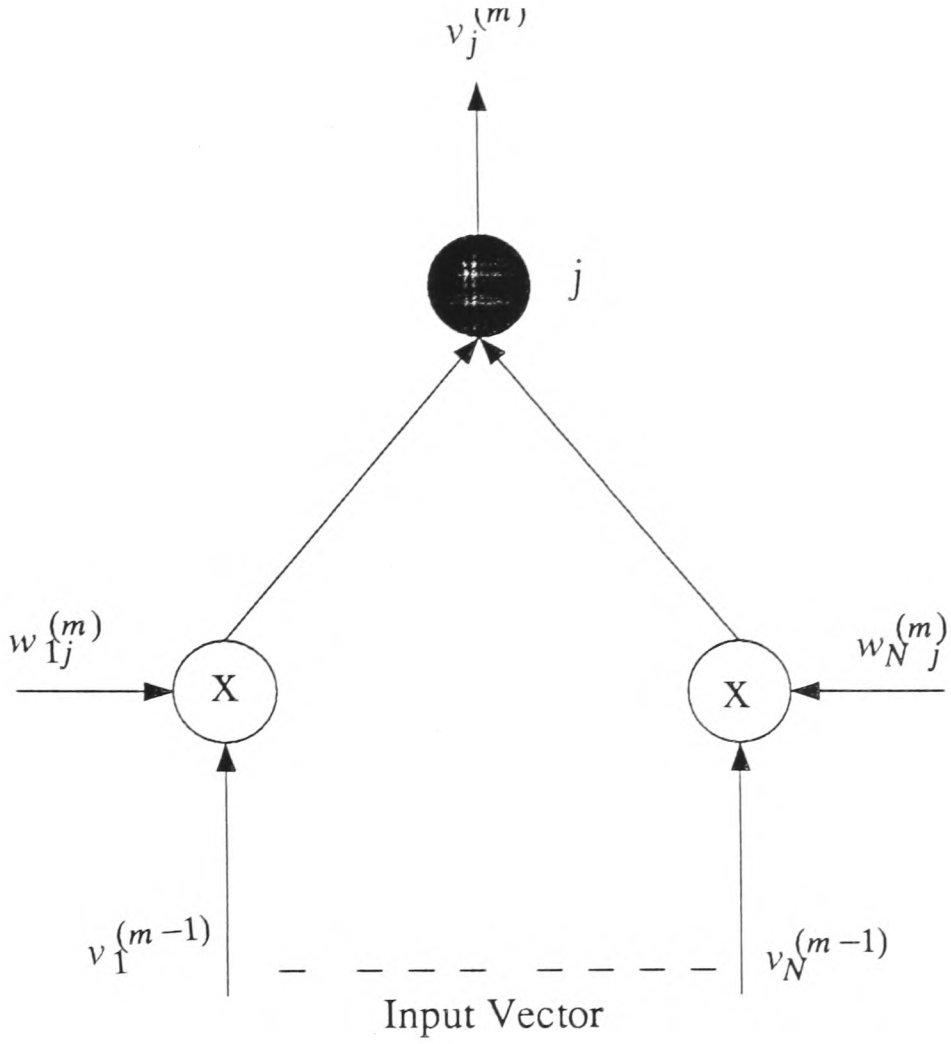
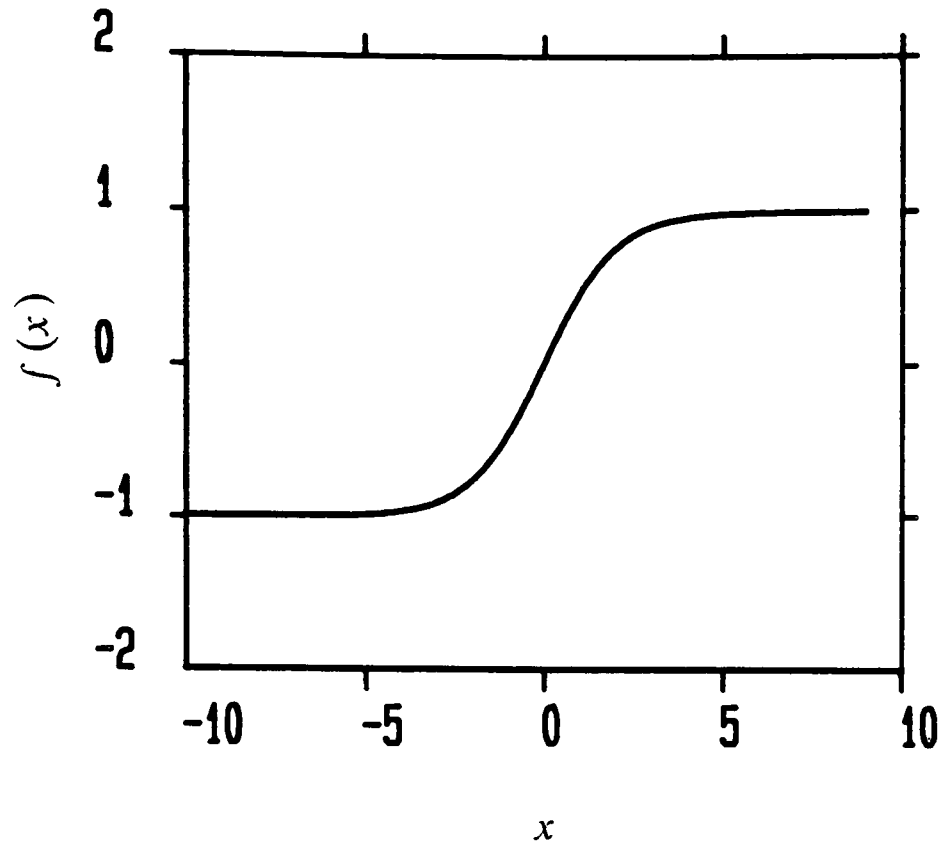
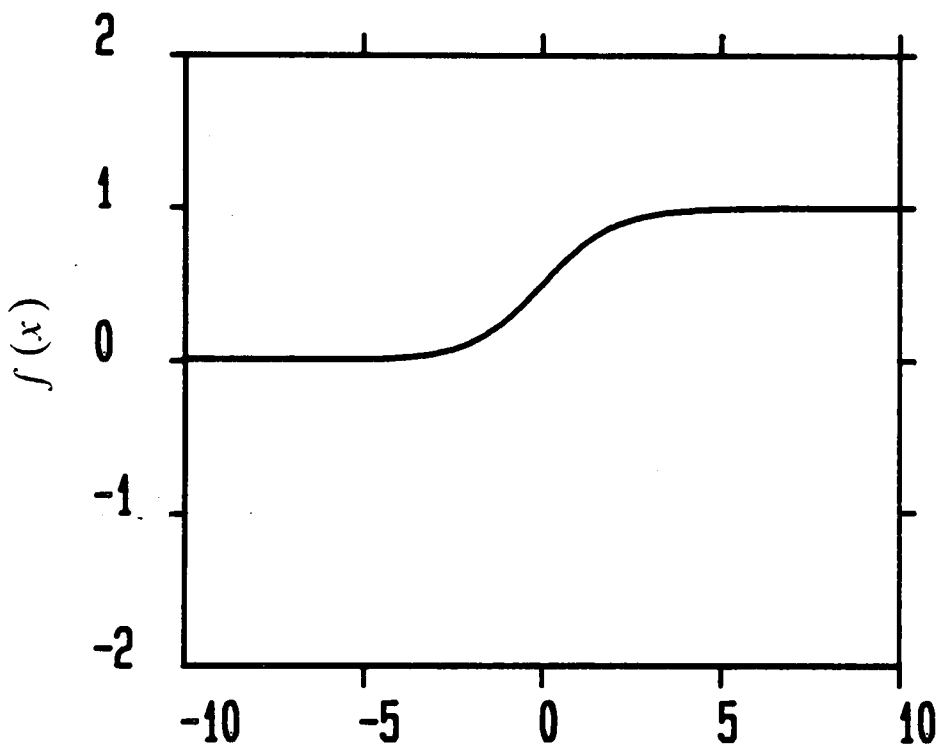


Figure (3.5) j^{th} neuron in m^{th} layer in multi-layer perceptrons.



(a)



(b)

Figure (3.6) Sigmoid type (a) $f(x) = \frac{1-e^{-x}}{1+e^{-x}}$ (b) $f(x) = \frac{1}{1+e^{-x}}$.

In the multi-layer perceptron [31],[45],[46] a number of neurons of the type described above are arranged in layers, as depicted in Figure (3.7) although there are generally different number of neurons in each layer. A multi-dimensional input is passed to each neuron in the first layer. The outputs of the first layer neurons then become inputs to the neurons in the second layer, and so on. The output of the network is therefore the outputs of the neurons lying in the final layer. Thus, weighted connections exist from a neuron to every neuron in the succeeding layer, but no connections exist between neurons in the same layer. The multi-layer perceptrons are capable of performing complex, nonlinear mappings between the input and the output.

3.3 On the decision regions of multi-layer perceptrons

We confine our discussion to the complexity of multi-layer perceptron decision region with hard limiting nonlinearities. Consider an input signal vector $V^{(0)}(v_1^{(0)}, v_2^{(0)} \ldots\ldots v_\phi^{(0)})$ in ϕ-dimensional space R^ϕ. Each neuron in the first hidden layer acts as a (ϕ-1) dimensional hyperplane with equation

$$\sum_{i=1}^{\phi} w_{ij}^{(1)} v_i^{(0)} + I_j^{(1)} = 0 \qquad (3.4)$$

that forms closed half-spaces. A polyhedral subset of R^ϕ is a finite intersection of closed half-spaces [47]. The polyhedral sets formed in the input space can be divided into bounded polyhedral sets and unbounded polyhedral sets. Bounded polyhedral sets are those inside which the input variables are bounded, while unbounded polyhedral sets comprise variables which can be unbounded.

In ϕ-dimensional space, the maximum number ($C(p, \phi)$) of polyhedral sets that are linearly separable using p hidden neurons was given by Schlafli [48], Makhoul et al.,

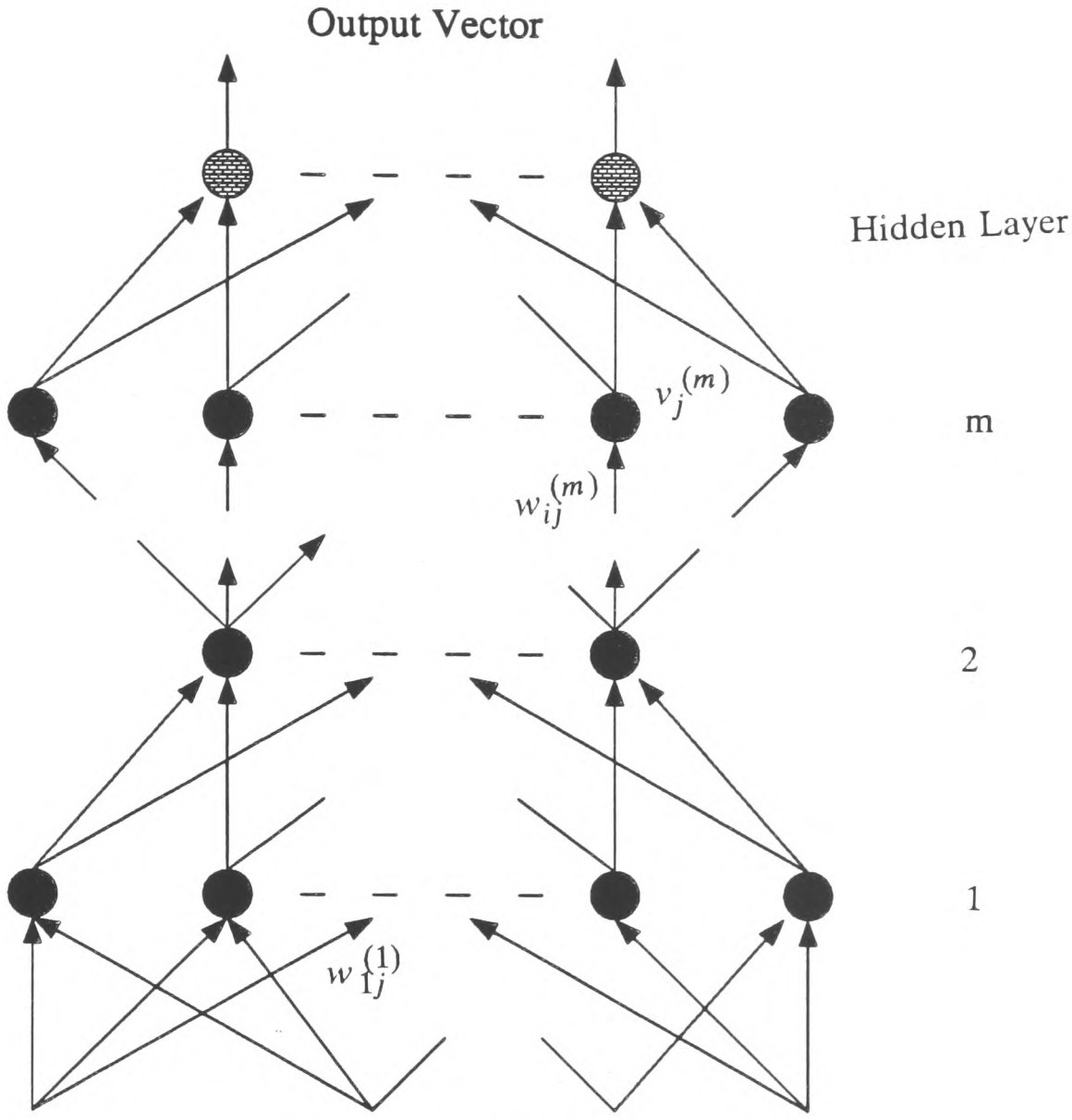


Figure (3.7) Multi-layer perceptron architecture
● : neuron in the hidden layer.

[49], and Mirchandani et al., [50], and the results are summarized here:

$$C(p,\phi) = \sum_{i=0}^{\min(p,\phi)} \binom{p}{i} = \begin{cases} 2^p & ;\ p \le \phi \\ \sum_{i=0}^{\phi} \binom{p}{i} & ;\ p > \phi \end{cases} \tag{3.5}$$

where

$$\binom{p}{i} = \frac{p!}{i!(p-i)!} \tag{3.6}$$

The number of unbounded polyhedral sets in $C(p,\phi)$ is

$$C_u(p,\phi) = \begin{cases} 2^p & ;\ p \le \phi \\ 2\sum_{i=0}^{\phi-1} \binom{p-1}{i} & ;\ p > \phi \end{cases} \tag{3.7}$$

and the number of bounded polyhedral sets in $C(p,\phi)$ is

$$C_b(p,\phi) = \begin{cases} 0 & ;\ p \le \phi \\ \binom{p-1}{\phi} & ;\ p > \phi \end{cases} \tag{3.8}$$

where $C(p,\phi) = C_b(p,\phi) + C_u(p,\phi)$.

In Table (3.1), we present a summary of the number of polyhedral sets, obtained by adding a dimension, a neuron or both [48],[49]. Based on the results of Table (3.1) and equation (3.5), the following properties are observed:

1. when we add a dimension while keeping the same number of neurons, this results in an increase the number of the polyhedral sets by $\binom{p}{\phi}$.

Table (3.1)

1	$C(p,\phi) = C(p,\phi-1) + \binom{p}{\phi}$
2	$C(p,\phi) = 2\,C(p-1,\phi) - \binom{p-1}{\phi}$
3	$C(p,\phi) = 2\,C(p-1,\phi-1) + \binom{p-1}{\phi}$

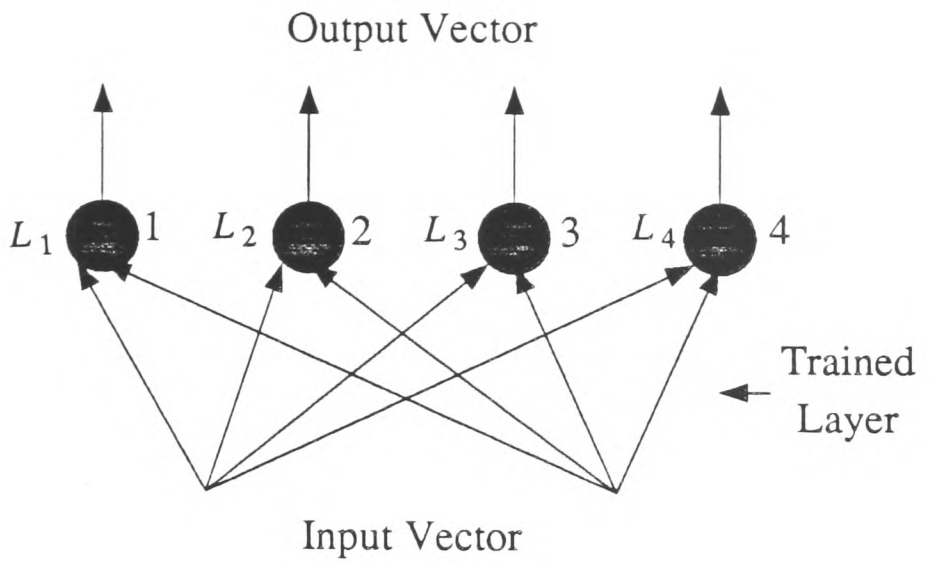
2 The number of polyhedral sets, $C(p,\phi)$ doubles every time we add a neuron, as long as $p<\phi$. However, for $p\geq\phi$, the result shows that on adding a single neuron, we have twice the previous number of polyhedral sets, $C(p-1,\phi)$ less $\binom{p-1}{\phi}$.

3. We double the number of polyhedral sets $C(p,\phi)$ plus $\binom{p-1}{\phi}$ every time we add a dimension and a neuron simultaneously.

Figure (3.8) shows the polyhedral sets formed by the first hidden layer with two inputs ($\phi=2$), and four neurons ($p=4$). Each polyhedral set is labeled with a binary number $L_1L_2L_3L_4$ that specifies the states of the neurons in the hidden layer. The four neurons in the hidden layer specifies four one-dimensional hyperplanes (ρ_i, i = 1,2,3,4) which divide the two-dimensional input space into eleven polyhedral sets. Among them three are bounded polyhedral sets, where the regions are { 0001, 0011, 1011 } and the remaining are unbounded polyhedral sets.

Using equations (3.5),(3.7) and (3.8), the number of polyhedral sets are $C(4,2)=\sum_{i=0}^{2}\binom{4}{i}=11$, the number of bounded polyhedral sets are $C_b(4,2)=\binom{3}{2}=3$ and the number of unbounded polyhedral sets are $C_u(4,2)=2\sum_{i=0}^{1}\binom{3}{i}=8$.

The polyhedral set forms the basic building block of the decision regions for the multi-layer perceptrons. For the two-layer perceptrons, the neuron in the second-layer (output layer) is used to group the polyhedral sets into decision regions. The output neuron with decision boundary



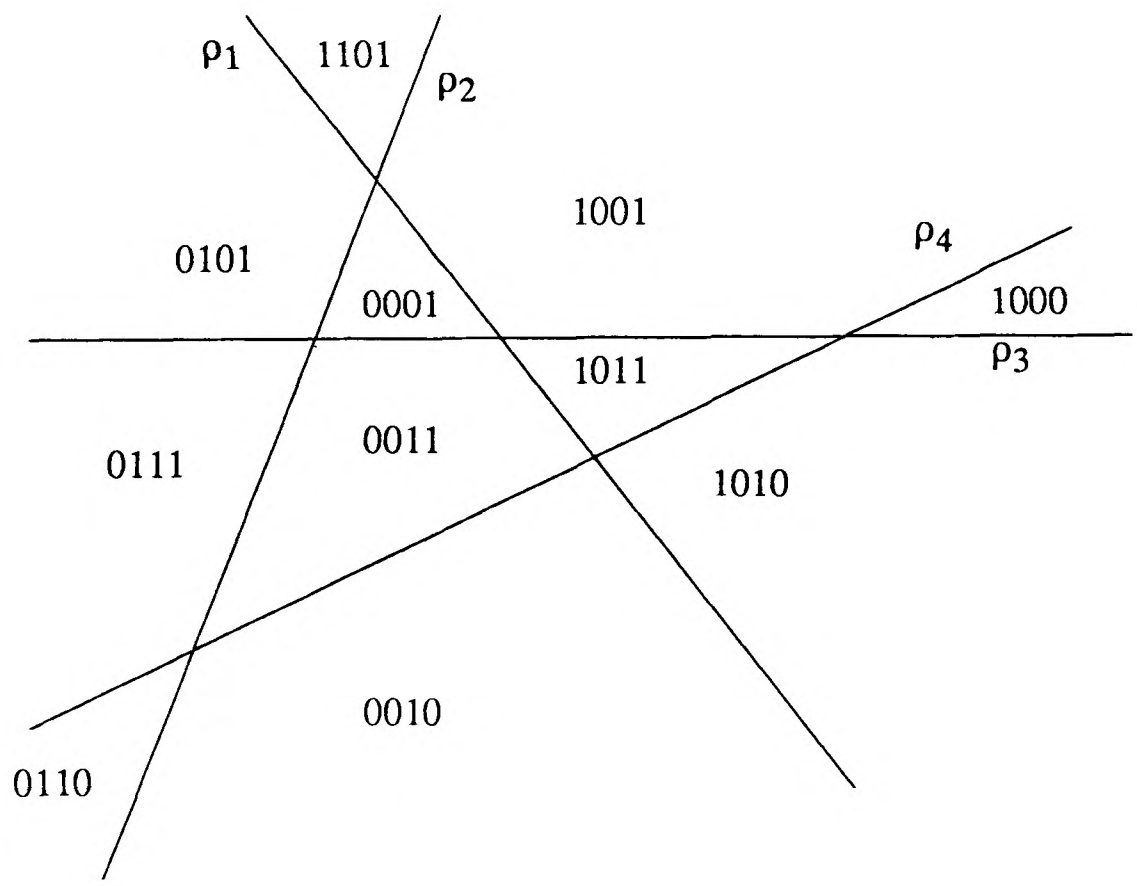


Figure (3.8) Polyhedral sets formed by the first hidden layer (two input nodes and four nodes in hidden layer).

$$-2L_1-2L_2+L_3+L_4=1.5 \tag{3.9}$$

will produce a convex (region 0011) decision region, as depicted in Figure (3.9), and decision boundary

$$-L_1-3L_2+2L_3+3L_4=2.5 \tag{3.10}$$

will produce a non-convex (regions 0001, 0011, 1011) decision region, as depicted in Figure (3.10), and decision boundary

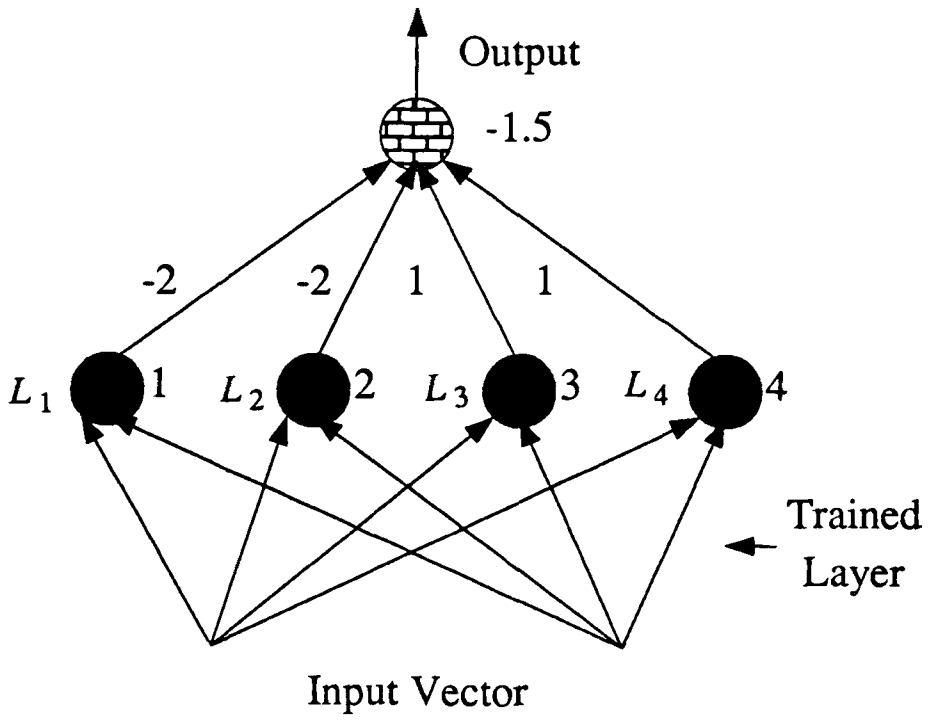
$$4L_1+5L_2-3L_3-4L_4=1.5 \tag{3.11}$$

will produce a disconnected (regions 1101, 0110, 1000) decision region, as shown in Figures (3.11).

The above results show that two-layer perceptrons are capable of forming convex, nonconvex and disconnected decision regions in the input space. The decision region is simply a finite union of polyhedral sets.

In the multi-layer perceptron for more than one hidden layer, each neuron in the second hidden layer groups the polyhedral sets which are produced in the first hidden layer to form an intermediate decision region, and so on. Ultimately the final layer groups the intermediate decision region in the layer next to it to form the output decision region. Since each of the intermediate decision regions is a finite union of polyhedral sets, so as the output decision region.

Summing up, a ϕ-dimensional input space is linearly separable into $C(p,\phi)$ polyhedral sets, and we may associate these polyhedral sets with classes. That is, the $C(p,\phi)$ polyhedral sets may be merged into F classes where $F \leq C(p,\phi)$. Furthermore, the number of separable polyhedral sets identifies the minimum number of



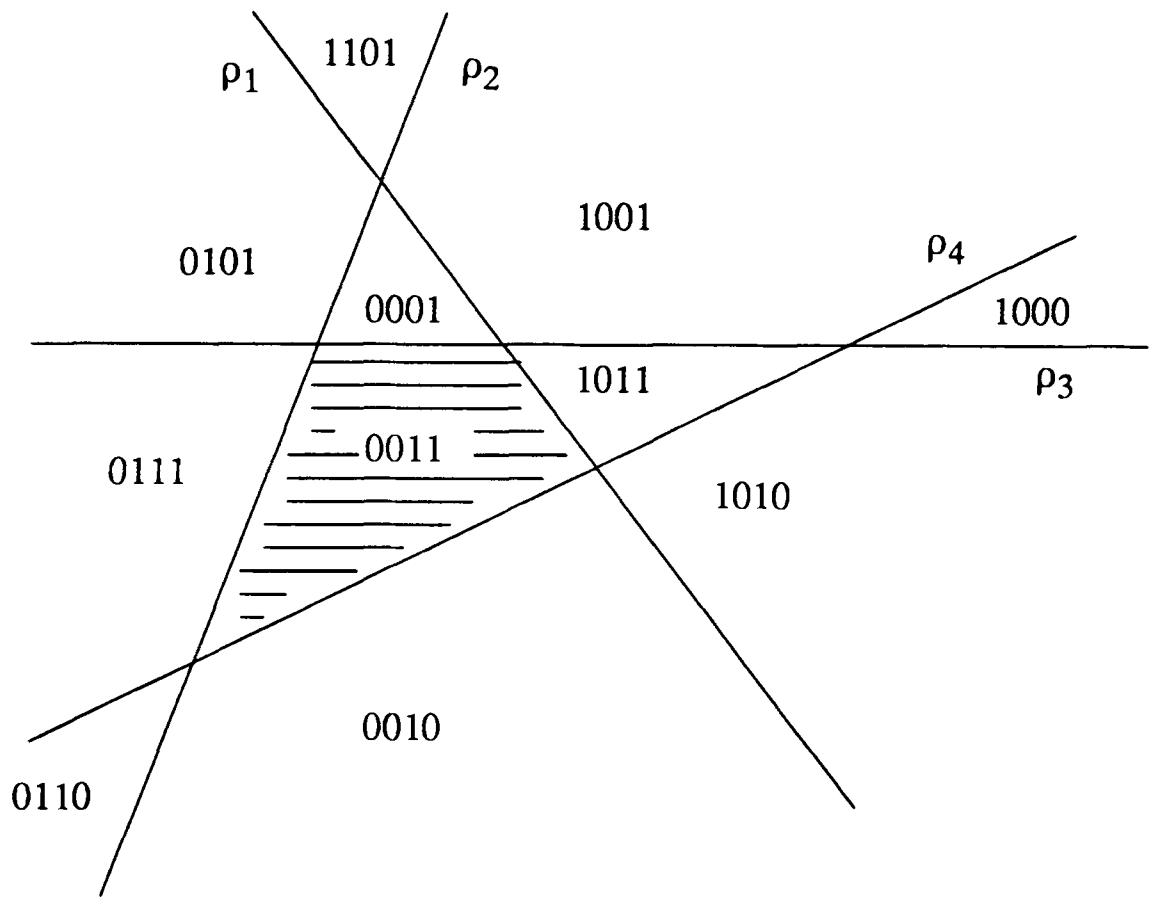
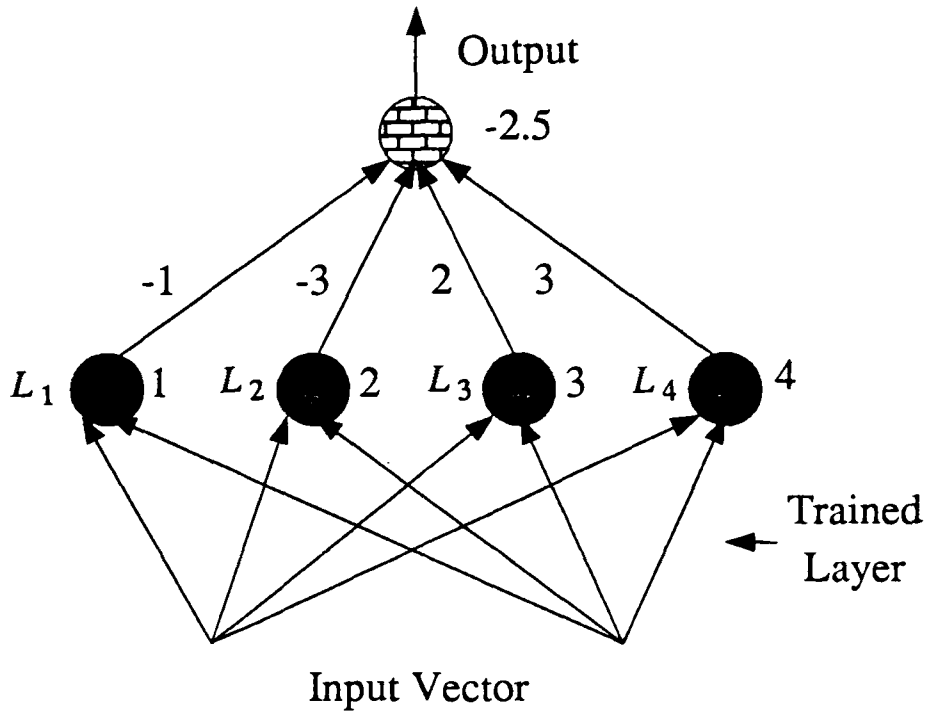


Figure (3.9) Convex decision regions formed by the two-layer perceptrons (two input nodes, four nodes in hidden layer and single output).

Shaded arca denotes decision region for " 1 ".



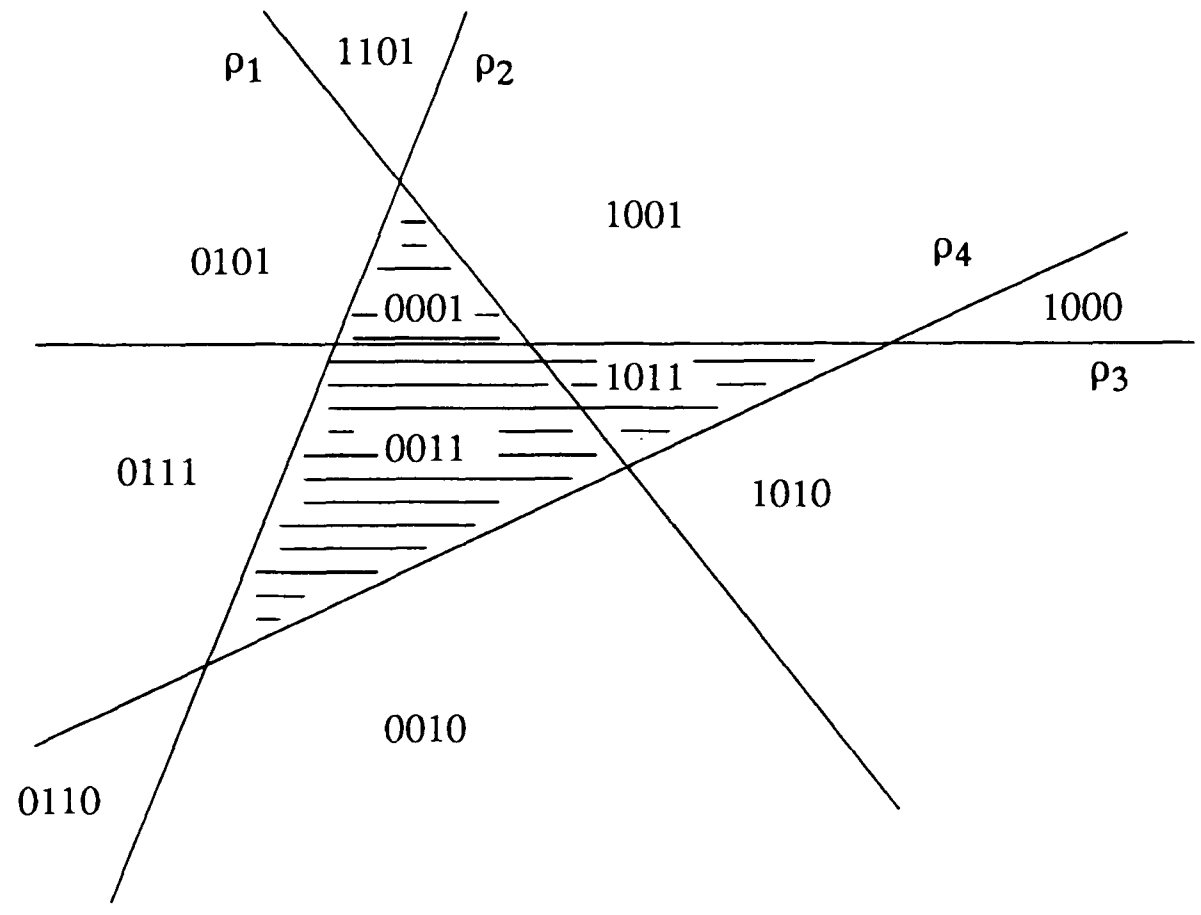
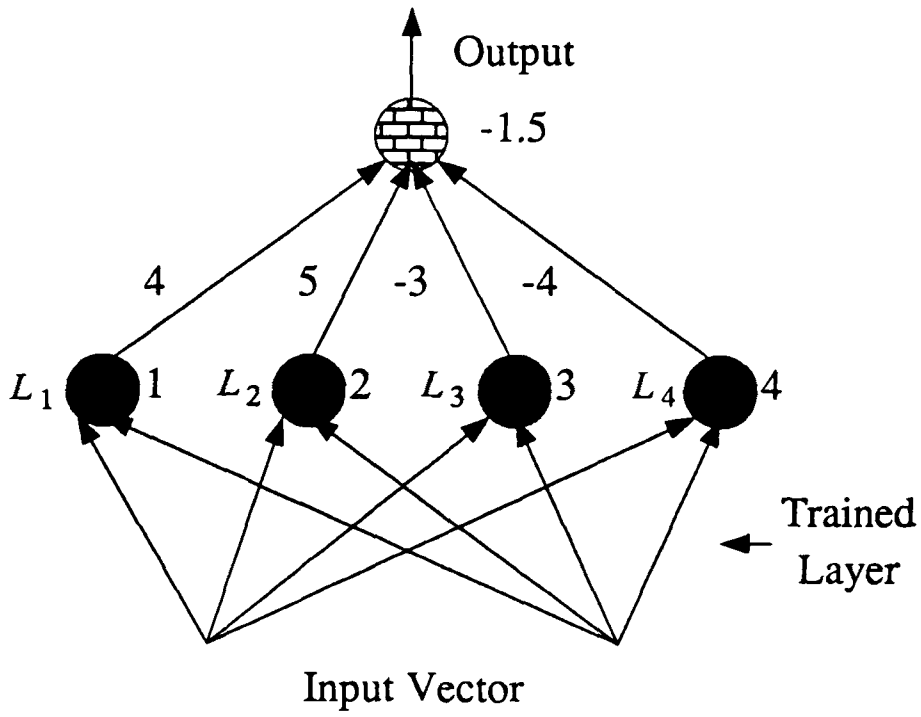


Figure (3.10) Nonconvex decision regions formed by the two-layer perceptrons (two input nodes, four nodes in hidden layer and single output).

Shaded area denotes decision region for " 1 ".



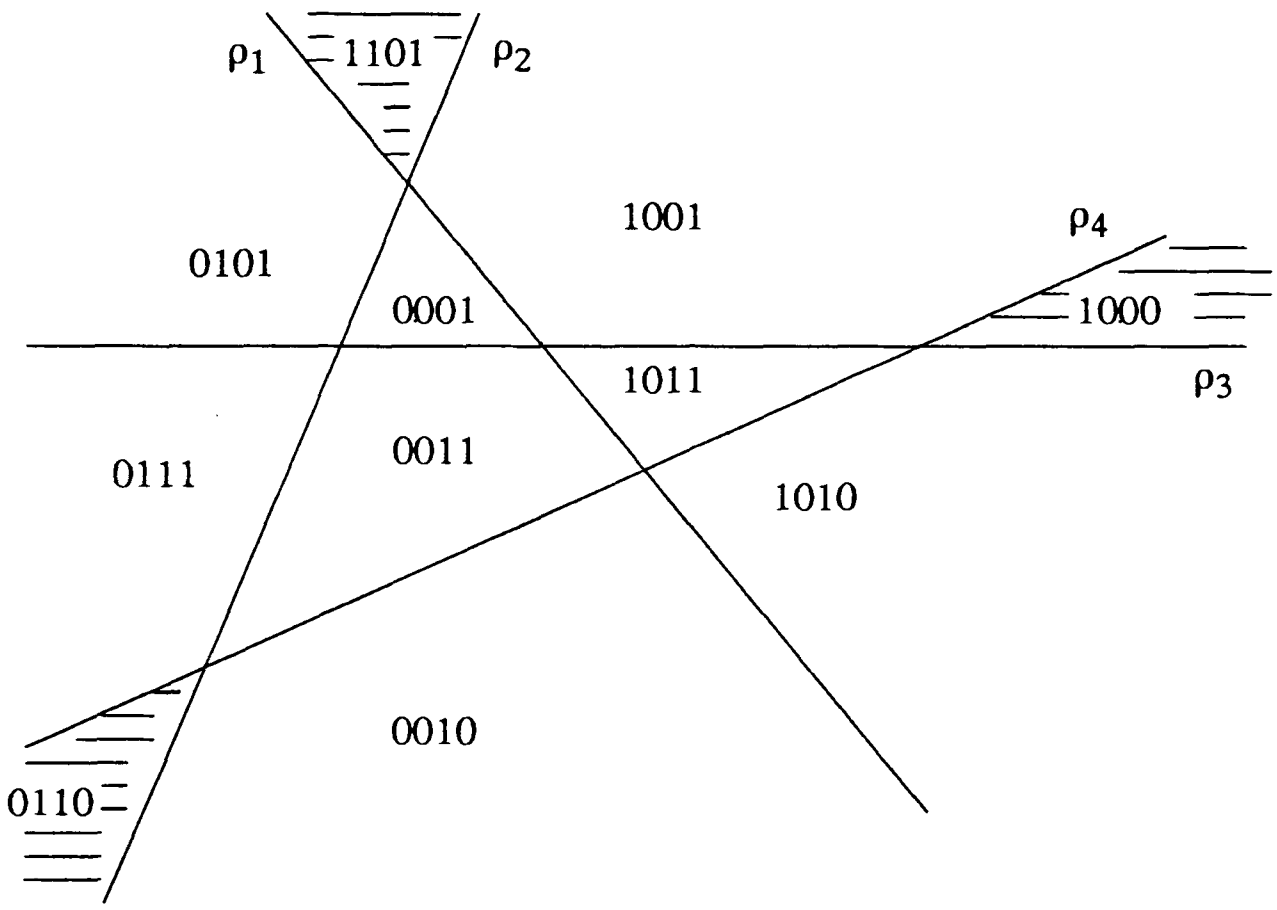


Figure (3.11) Disconnected decision regions formed by the two-layer perceptrons (two input nodes, four nodes in hidden layer and single output).

Shaded area denotes decision region for " 1 ".

input training samples. The main limitation of the multi-layer perceptron with the hard limiting nonlinearity is that the decision regions are bounded by straight line segments (hyperplanes). Even though we only dealt with hard limiting nonlinearities, the same general conclusions apply to sigmoidal nonlinearities in the networks. The behaviour of the networks with the sigmoid function is more complex because the decision regions are typically bounded by smooth curves instead of by straight line segments.

The Figures (3.12), and (3.13) show the decision regions formed by a three-layer perceptrons using sigmoid function (equation (3.3)) employed to equalize two different distorted channels (Table (3.2)). It can be seen that the decision regions formed by the multi-layer perceptrons are complex and have highly nonlinear boundaries.

3.4 Learning algorithms- Back propagation

An iterative learning algorithm, dubbed back propagation enabling multi-layer perceptrons to learn more sophisticated tasks than before, was suggested by Rumelhart et al (1986) [31]. The learning procedure involves 2 phases. During the first phase the input is recoded into an internal representation (hidden layer) and the final output is generated by the combinational operations on the internal representation. This output value is then compared with the desired output, resulting in an error signal. The second phase involves the propagation of the error signal back through the network to update the weights using the stochastic gradient algorithm.

From equation (3.1), the output value of j^{th} neuron in the m^{th} layer can be rewritten as

$$v_j^{(m)} = f_j(x_j^{(m)}) \tag{3.12}$$

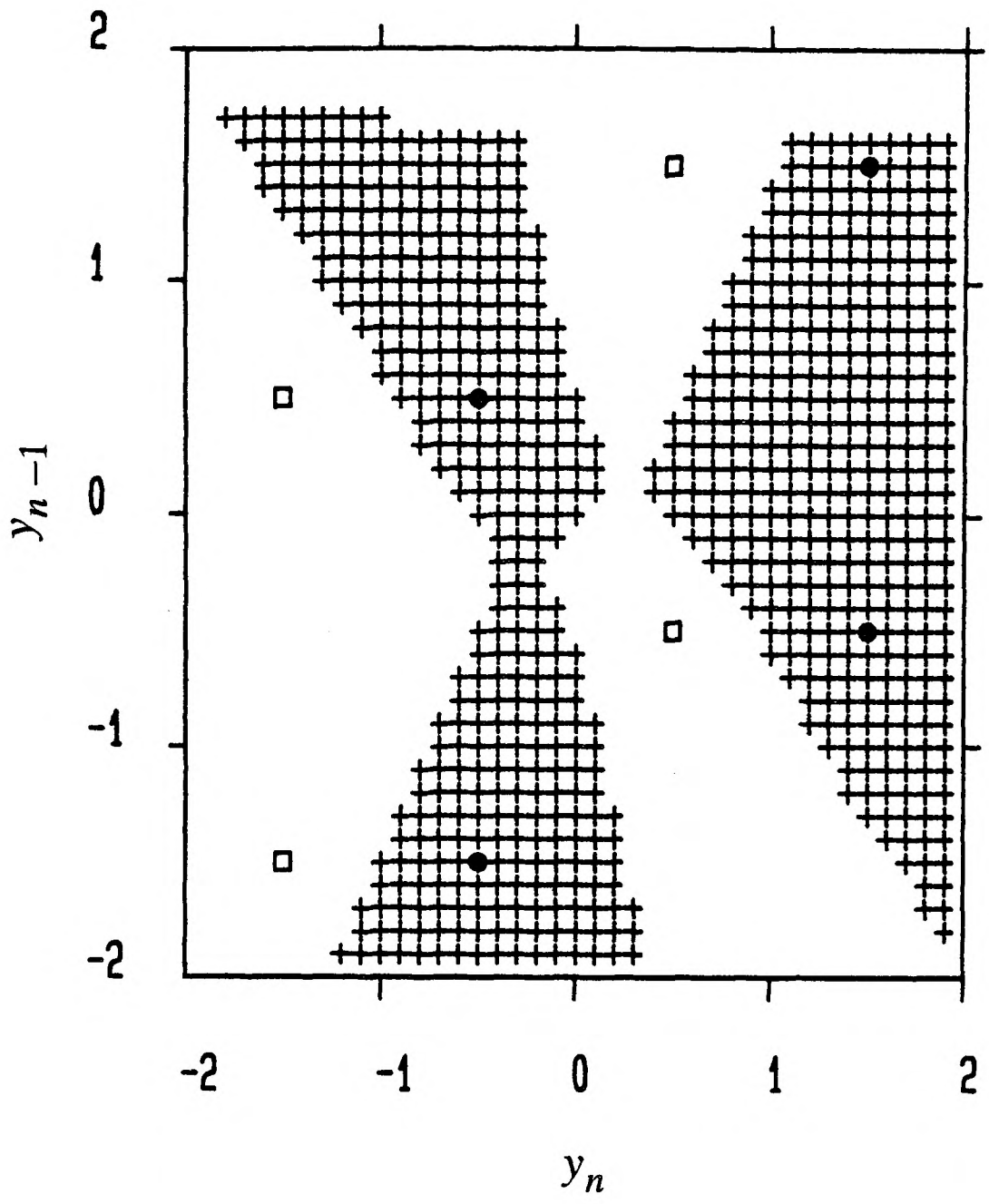


Figure (3.12) The decision regions formed by the three-layer perceptrons (two input nodes, nine nodes in hidden layer one, three nodes in hidden layer 2, and single output) employed to equalize the distorted channel 1 in Table (3.2), following a training period of 1000 samples duration, where the noise variance σ_v^2 was 0.01.

o : signal vector belonging to " 1 ",

□ : signal vector belonging to " -1 ".

Shaded area denotes decision region for " 1 ".

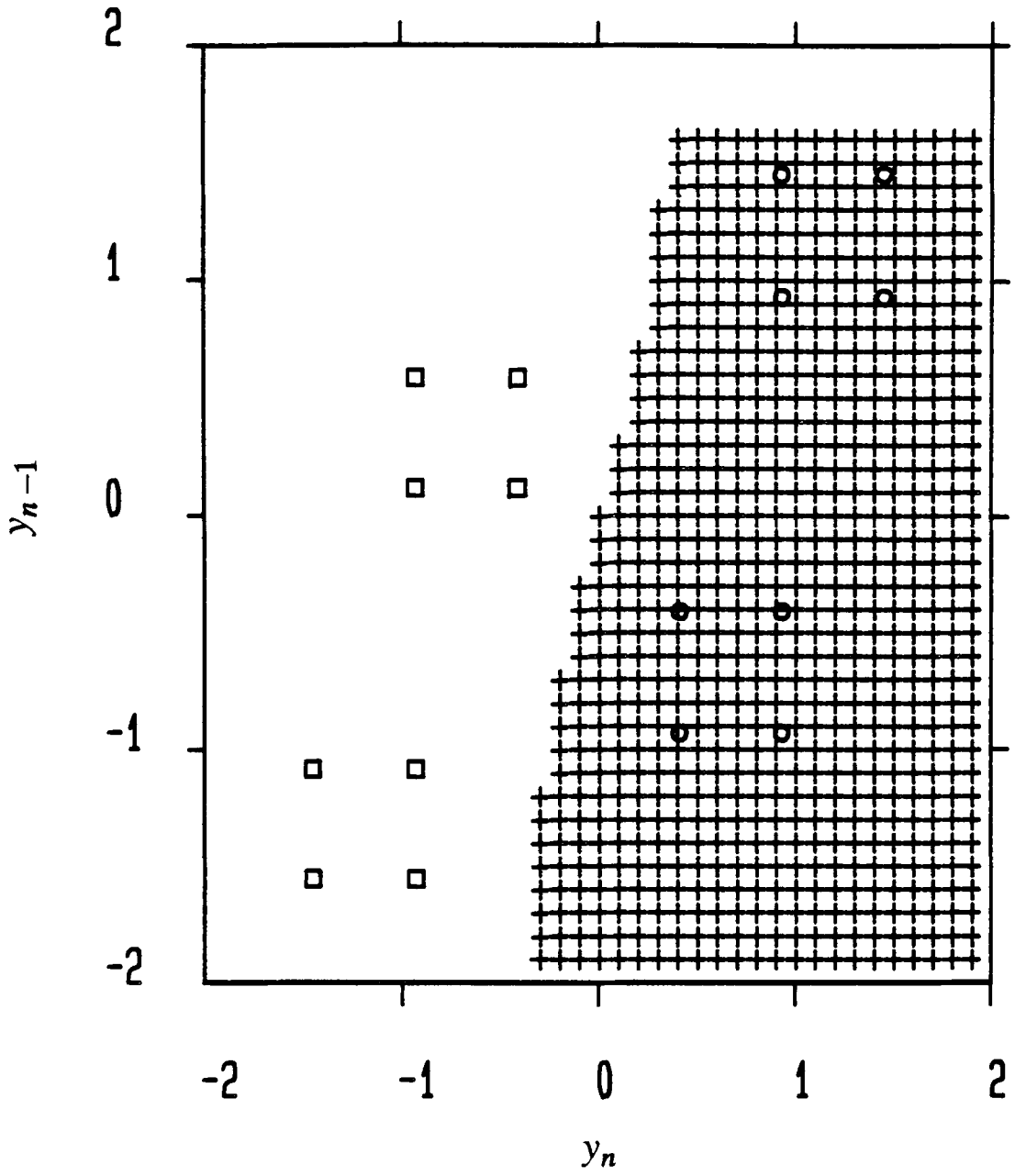


Figure (3.13) The decision regions formed by the three-layer perceptrons (two input nodes, nine nodes in first hidden layer, three nodes in second hidden layer, and single output) employed to equalize the distorted channel 2 in Table (3.2) with one bit delay ($d = 1$), following a training period of 1000 samples duration, where the noise variance σ_v^2 was 0.09.

o : signal vector belonging to " 1 ",

□ : signal vector belonging to " -1 ".

Shaded area denotes decision region for " 1 ".

Table (3.2) Channel Impulse Responses

Channel No.	Impulse Response	type
1	$0.5 + 1.0z^{-1} + 0.0z^{-2}$	Non-minimum phase
2	$0.2602 + 0.9298z^{-1} + 0.2602z^{-2}$	Non-minimum phase

where $x_j^{(m)} = \sum_i w_{ij}^{(m)} v_j^{(m-1)} + I_j^{(m)}$ and layer $m = \{1,2,\ldots. M\}$.

Given an input $V^{(0)} \in R^N$ together with an associated desired output $t_j \in R$. The aim is to vary the weights $\{ w_{ij}^{(m)} \}$ so as to minimize the mean square error E between the final output and the desired output, where

$$E = \frac{1}{2} \sum_j (t_j - v_j^{(M)})^2 \tag{3.13}$$

where M denotes the final output layer.

A learning rule for $w_{ij}^{(m)}$ and $I_j^{(m)}$ were developed by performing gradient descent on E, i.e.

$$\Delta w_{ij}^{(m)} = -\eta \frac{\partial E}{\partial w_{ij}^{(m)}} \tag{3.14}$$

$$\Delta I_j^{(m)} = -\beta \frac{\partial E}{\partial I_j^{(m)}} \tag{3.15}$$

where

$$-\frac{\partial E}{\partial w_{ij}^{(m)}} = \delta_j^{(m)} v_j^{(m)} \tag{3.16}$$

$$-\frac{\partial E}{\partial x_j^{(m)}} = \delta_j^{(m)} \tag{3.17}$$

where η is the learning gain and β is the adaptation gain of threshold level.

The error signal $\delta_j^{(m)}$ for layer m is calculated starting from the output layer M

$$\begin{aligned} \delta_j^{(M)} &= -\frac{\partial E}{\partial x_j^{(M)}} = -\frac{\partial E}{\partial v_j^{(M)}} \frac{\partial v_j^{(M)}}{\partial x_j^{(M)}} \\ &= (t_j - v_j^{(M)}) f_j'(x_j^{(M)}) \end{aligned} \tag{3.18}$$

and recursively back-propagating the error signal to lower layers

$$\begin{aligned}\delta_j^{(m)} &= -\frac{\partial E}{\partial x_j^{(m)}} = -\frac{\partial E}{\partial v_j^{(m)}}\frac{\partial v_j^{(m)}}{\partial x_j^{(m)}} \\ &= -\sum_l \frac{\partial E}{\partial x_l^{(m+1)}}\frac{\partial x_l^{(m+1)}}{\partial v_j^{(m)}} f_j'(x_j^{(m)}) \\ &= f_j'(x_j^{(m)})\sum_l \delta_l^{(m+1)} w_{lj}^{(m+1)}\end{aligned} \tag{3.19}$$

where $m \in [1,2...M-1]$, $f'(\cdot)$ is the derivative of $f(\cdot)$, and l is over all neurons in the layer above neuron j.

$\Delta I_j^{(m)}$ in equation (3.15) can be expressed in terms of $\delta_j^{(m)}$, where

$$\begin{aligned}\frac{\partial E}{\partial I_j^{(M)}} &= \frac{\partial E}{\partial v_j^{(M)}}\frac{\partial v_j^{(M)}}{\partial I_j^{(M)}} \\ &= -(t_j - v_j^{(M)}) f_j'(x_j^{(M)}) = -\delta_j^{(M)}\end{aligned} \tag{3.20}$$

and

$$\begin{aligned}\frac{\partial E}{\partial I_j^{(m)}} &= \sum_l \frac{\partial E}{\partial x_l^{(m+1)}}\frac{x_l^{(m+1)}}{\partial I_j^{(m)}} \\ &= -f_j'(x_j^{(m)}) \sum_l \delta_l^{(m+1)} w_{lj}^{(m+1)} = -\delta_j^{(m)}\end{aligned} \tag{3.21}$$

The updating of weights and threshold levels of the m^{th} layer can be accomplished by the following rules.

$$\Delta w_{ij}^{(m)}(n+1) = \eta \delta_j^{(m)}(n)\, v_j^{(m-1)}(n) + \alpha \Delta w_{ij}^{(m)}(n) \tag{3.22}$$

and

$$\Delta I_j^{(m)}(n+1) = \beta \delta_j^{(m)}(n) \tag{3.23}$$

Where $\alpha(0 \le \alpha < 1)$ is the momentum parameter and $m \in [1,2....M]$. Substituting equations (3.2) and (3.3) in equations (3.20) and (3.21), we obtain the error signal, $\delta_j^{(m)}$ and the results are summarized in Tables (3.3) and (3.4).

Table (3.3) Error Feedback Signal

Sigmoid type	$f(x)=\frac{1}{1+e^{-x}}$
Output layer	$\delta_j^{(M)}=(t_j-v_j^{(M)})v_j^{(M)}(1-v_j^{(M)})$
Hidden layer	$\delta_j^{(m)}=v_j^{(m)}(1-v_j^{(m)})\sum_l \delta_l^{(m+1)}w_{lj}^{(m+1)}$ $m \in [1,2,....M-1]$

Table (3.4) Error Feedback Signal

Sigmoid type	$f(x)=\frac{1-e^{-x}}{1+e^{-x}}$
Output layer	$\delta_j^{(M)}=(t_j-v_j^{(M)})(1-v_j^{2\,(M)})/2$
Hidden layer	$\delta_j^{(m)}=(1-v_j^{2\,(m)})\sum_l \delta_l^{(m+1)}w_{lj}^{(m+1)}/2$ $m \in [1,2.....M-1]$

The learning gain, η , is a constant which scales the adjustment to the weights so that the optimum values can be approached more or less quickly. To allow rapid learning a momentum term, $\Delta w_{ij}^{(m)}(n)$, scaled by α is used to filter out high frequency variation of the weight vector. As a result, the convergence rate is much faster and the weight changes are smoothed.

The $w's$ (weights) and $I's$ (threshold levels) in equations (3.22) and (3.23) are values specified by the training algorithm.

3.5 Conclusions

A relationship has been established between the maximum number of polyhedral sets ($C(p,\phi)$), the number of neurons in first the hidden layer (p), and the dimension of the input space (ϕ). The polyhedral sets can be associated with classes. That is, the $C(p,\phi)$) polyhedral sets may be merged into F classes, where $F \leq C(p,\phi)$. Furthermore, the number of polyhedral sets identifies the minimum number of input training samples.

In the single-layer perceptron, that is a perceptron composed of a single neuron, the decision regions are delimited by the hyperplanes. However in the multi-layer perceptrons we are able to group the polyhedral sets to form considerably more complex decision regions including convex, non-convex, and disconnected regions. Furthermore, multi-layer perceptrons using the sigmoid function have highly nonlinear decision boundaries. It is this latter capability of multi-layer perceptrons which will prove invaluable when we apply them to the problem of channel equalization.

CHAPTER 4

A PERCEPTRON - BASED EQUALIZER

4.1 Introduction

The purpose of using an equalizer in the receiver of a digital communications system is to reduce the ISI as much as possible in order to maximize the probability of correct decisions. Usually, the structure of equalizers is based on the linear filter. This is known as linear equalizer. The decision regions of a linear equalizer are always limited by hyperplanes. The linearity of these decision boundaries limit the performance of the system for example data rate and bit error rate. Nonlinear adaptive equalizers give superior performance (higher resolution) to linear equalizers, especially when signal to noise conditions are poor, but are more complex to implement. Recent advances in computer and VLSI technologies, however, permit signal processing systems to handle just such sophisticated and computationally complex algorithms. Consequently research into nonlinear adaptive equalizers has new-found practical and commercial importance.

In this chapter, we discuss a new approach for nonlinear equalization using the multilayer perceptron architecture. The performance of the perceptron-based equalizer and some of its properties are examined. The performance of the perceptron-based equal-

izer, in terms of the convergence time and the bit error rate, was compared to the performance of the conventional LMS linear transversal equalizer(LMS linear equalizer). Comparisons of the performance of this perceptron-based equalizer with the optimal linear equalizer and the optimal equalizer based on the statistical results in chapter 2 are also provided.

4.2 Perceptron - Based Equalizer

4.2.1 Architecture and Modelling

A perceptron-based equalizer structure [51],[52], consists of a number of neurons which are arranged in layers, as shown in Figure (4.1). The three-layer perceptron (two hidden layers, and output layer) is sufficient for channel equalization. The reason is that three-layer structures are sufficient to realise arbitrary complex, nonlinear decision regions [53],[54],[55] for performing nonlinear mappings between the input and the output. It is this capability of the multi-layer perceptron which will prove invaluable as a channel equalizer in digital communications.

For simplicity the short hand notation { $(N_1, N_2, N_3,.., N_M)$ MLP structure with order N } will be used to indicate that the number of received signal samples is N, the number of neurons in hidden layer 1(H_1) is N_1, the number of neurons in hidden layer 2(H_2) is N_2 , and so on, for the perceptron-based equalizer.

The input to the network (feedforward filter) is the sequence of noisy received signal samples { y_n } from the dispersion channel. At time n, the input $N \times 1$ received signal vector

$$V^{(0)} = [\, y_n, y_{n-1} \cdots\cdots y_{n-N+1} \,]^T \qquad (4.1)$$

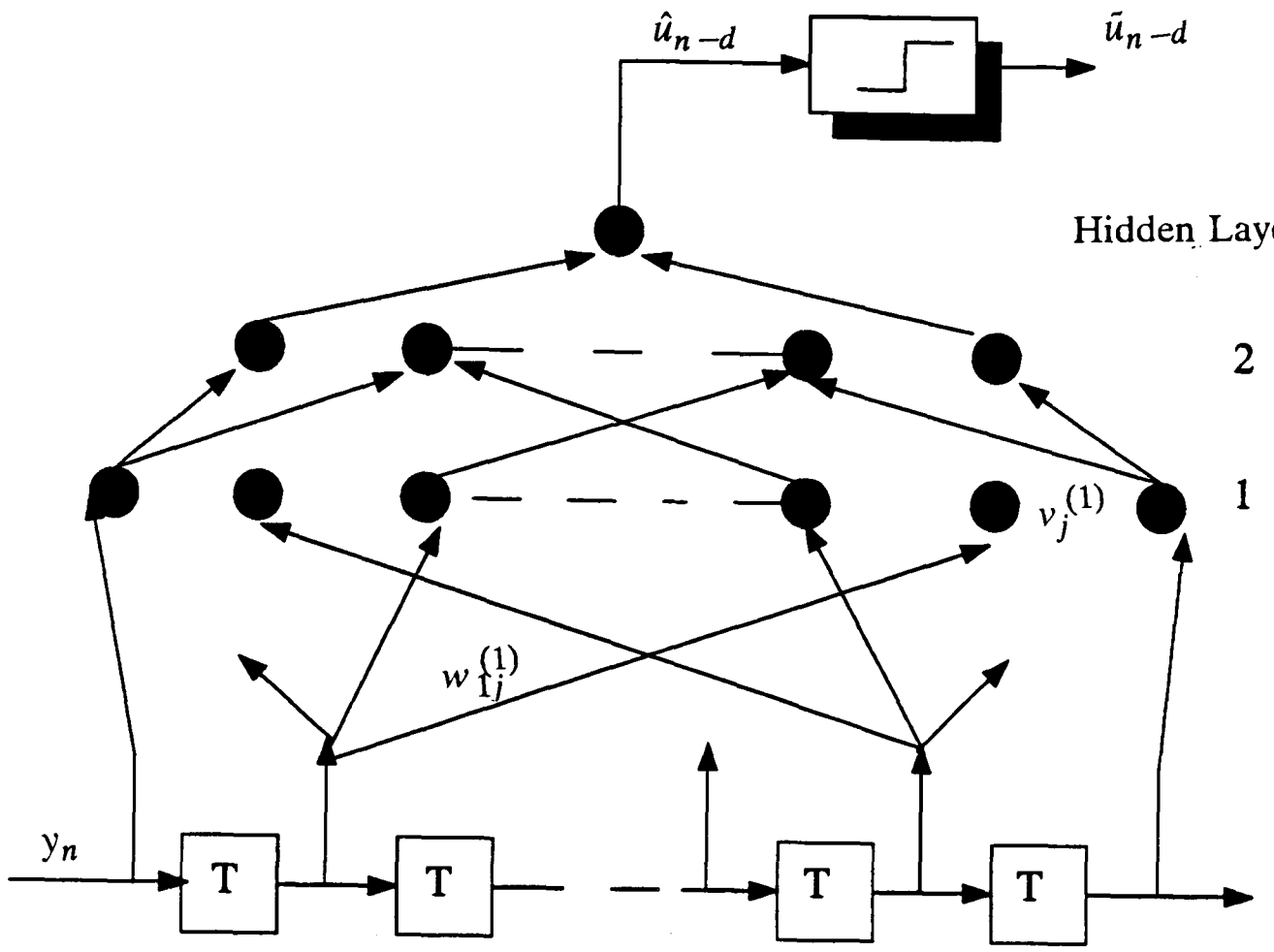


Figure (4.1) Perceptron-based equalizer architecture

is in the feedforward filter.

The $N_1 \times 1$ vector in the output of hidden layer 1 is :

$$V^{(1)} = \left[v_1^{(1)} \; v_2^{(1)} \; . \; . \; v_j^{(1)} \; . \; . \; v_{N_1}^{(1)} \right]^T \tag{4.2}$$

where

$$v_j^{(1)} = f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} \, y_{n-i} + I_j^{(1)} \right) \qquad j = 1,2,.....N_1 \tag{4.3}$$

The $N_2 \times 1$ vector in the output of hidden layer 2 is :

$$V^{(2)} = \left[v_1^{(2)}, \; v_2^{(2)}, \; . \; . \; v_k^{(2)}, \; . \; . \; v_{N_2}^{(2)} \right]^T \tag{4.4}$$

where

$$v_k^{(2)} = f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} \, v_j^{(1)} + I_k^{(2)} \right) \qquad k = 1,2,.....N_2 \tag{4.5}$$

and so on.

The final output of the three-layer perceptron is

$$v_o^{(3)} = \hat{u}_{n-d} = f_o\left(\sum_{k=1}^{N_2} w_{ko}^{(3)} \, v_k^{(2)} + I_o^{(3)} \right) \tag{4.6}$$

Where $\hat{u}_{n-d}$ is the estimated signal at time n, and d is a delay parameter. Substituting equations (4.3) and (4.5) into equation (4.6), yields

$$\hat{u}_{n-d} = f_o\left(\sum_{k=1}^{N_2} w_{ko}^{(3)} \, f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} \, f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} \, y_{n-i} + I_j^{(1)} \right) + I_k^{(2)} \right) + I_o^{(3)} \right) \tag{4.7}$$

The nonlinear detector can be modelled as a threshold function $g(x)$ and is defined as :

$$g(\hat{u}_{n-d}) = \tilde{u}_{n-d} = \begin{cases} 1 & \text{if } \hat{u}_{n-d} \geq 0 \\ -1 & otherwise \end{cases} \tag{4.8}$$

The $w's$ (weights) and $I's$ (threshold levels) in equation (4.7) are values specified by the training algorithm, so that after training is finished the equalizer will self-adapt to changes in channel characteristics occurring during transmission (Decision-Directed mode).

The number of total weights (L_t) for a fully connected multi-layer perceptron $(f : R^N \rightarrow R^{N_M})$ is

$$L_t = NN_1 + N_1N_2, \ldots, + N_{M-1}N_M = NN_1 + \sum_{i=1}^{i=M-1} N_iN_{i+1} \tag{4.9}$$

whereas in linear transversal equalizer, the total number of weights is N (order).

4.2.2 Learning algorithm

The learning algorithm of the perceptron-based equalizer is based the back propagation. The increments used in updating of the weights, Δw_{ij} and threshold levels, ΔI_j of the m^{th} layer can be accomplished by the following rules (see equations (3.22) and (3.23)).

$$\Delta w_{ij}^{(m)}(n+1) = \eta \delta_j^{(m)}(n)\, v_j^{(m-1)}(n) + \alpha \Delta w_{ij}^{(m)}(n) \tag{4.10}$$

and

$$\Delta I_j^{(m)}(n+1) = \beta \delta_j^{(m)}(n) \tag{4.11}$$

where η is the learning gain; α is the momentum parameter; β is the threshold level adaptation gain; and layer $m \in [1,2.....M]$.

The error signal $\delta_j^{(m)}$ for layer m is calculated starting from the output layer M

$$\delta_j^{(M)} = (t_j - v_j^{(M)})(1 - v_j^{2\,(M)})/2 \tag{4.12}$$

and recursively back-propagating the error signal to lower layers

$$\delta_j^{(m)} = (1 - v_j^{2\,(m)}) \sum_l \delta_l^{(m+1)} w_{lj}^{(m+1)}/2 \tag{4.13}$$

where $m \in [1,2...M-1]$, l is over all neurons in the layer above neuron j and t_j is the desired output. The results in equations (4.12) and (4.13) are based on the sigmoid function $f(x) = \frac{1-e^{-x}}{1+e^{-x}}$ (see equation (3.3)).

To allow rapid learning a momentum term, $\Delta w_{ij}^{(m)}(n)$, scaled by α is used to filter out high frequency variation of the weight vector. As a result, the convergence rate is much faster and the weight changes are smoothed.

4.3 Perceptron-Based Equalizer Performance on A Simulated Channel

The channel model used in the performance evaluation is given in z- transform notation by :

$$H(z) = 0.3482 + 0.8704\, z^{-1} + 0.3482\, z^{-2} \tag{4.14}$$

The digital message applied to the channel was in random bipolar form { -1,1 }. The channel output is corrupted by zero mean white Gaussian noise. For mathematical convenience, we normalize the received signal power signal to unity. Then the received signal to noise ratio (SNR) is simply the reciprocal of the noise variance at the input of the equalizer.

The performance was determined by taking an average of 800 individual runs, and the final bit error rate after convergence was measured. Each run had a different

random sequence and random starting weights. The sigmoid function in equation (3.3) was chosen for performance simulation.

4.3.1 Properties of Decision Regions in Perceptron-Based Equalizer

The bit error rate (BER) and the mean squared error (MSE), are the two most useful measures of equalizer performance. However, the varying decision boundaries (equation (2.19)), and (equation (2.28)) of equalizers are used to maximize the correct decisions. The decision boundary characteristics of the perceptron-based equalizer, in terms of SNR, was compared to both the decision boundaries of the optimal equalizer and the optimal linear equalizer. These comparisons provide a comprehensive basis for evaluation of the performance behaviour of the perceptron-based equalizer, in terms of SNR.

Figure 4.2 shows that the decision region formed by the perceptron-based equalizer { (9,3,1)MLP structure with order 2 } and the decision boundary formed by the optimal equalizer based on the maximum a-Posteriori (MAP) criterion (equation (2.19)), in the case of 10 dB SNR. The Maximum a-Posteriori criterion will yield a minimum probability-of error decision. It can be seen that the decision region formed by the perceptron is near that optimal decision region which suggests that the perceptron is utilising the available information in a way approaching maximum efficiency [52].

Furthermore, the decision boundary generated by the perceptron-based equalizer as a function of SNR is examined. When the SNR is low, it forms a nonlinear decision boundary as shown in Figure (4.3(a)). However as the SNR increases, the resulting decision boundary becomes increasingly linear, as shown in Figure (4.3(b)). Because the perceptron will receive very few input samples which are close to the optimal deci-

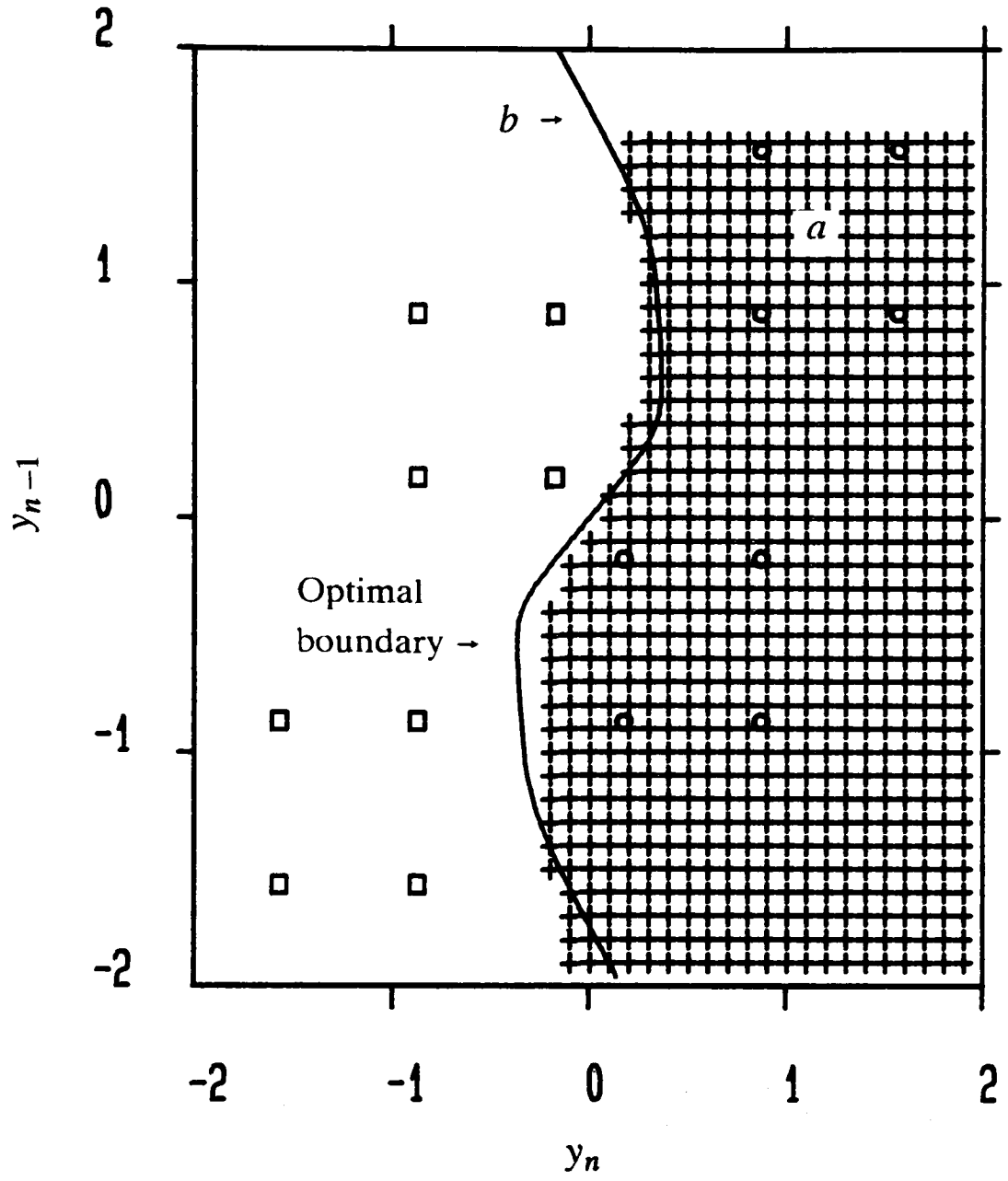


Figure (4.2) Decision region formed for SNR = 10 dB by

(a) perceptron-based equalizer { (9,3,1)MLP structure with $N=2$ } after a training period of 1000 samples duration: $\eta = 0.3$, $\alpha = 0.3$ and $\beta = 0.05$. Shaded area denotes decision region for " 1 ".

(b) Optimal decision boundary formed by the MAP criterion.

o : signal vector belonging to " 1 ",

□ : signal vector belonging to " -1 ".

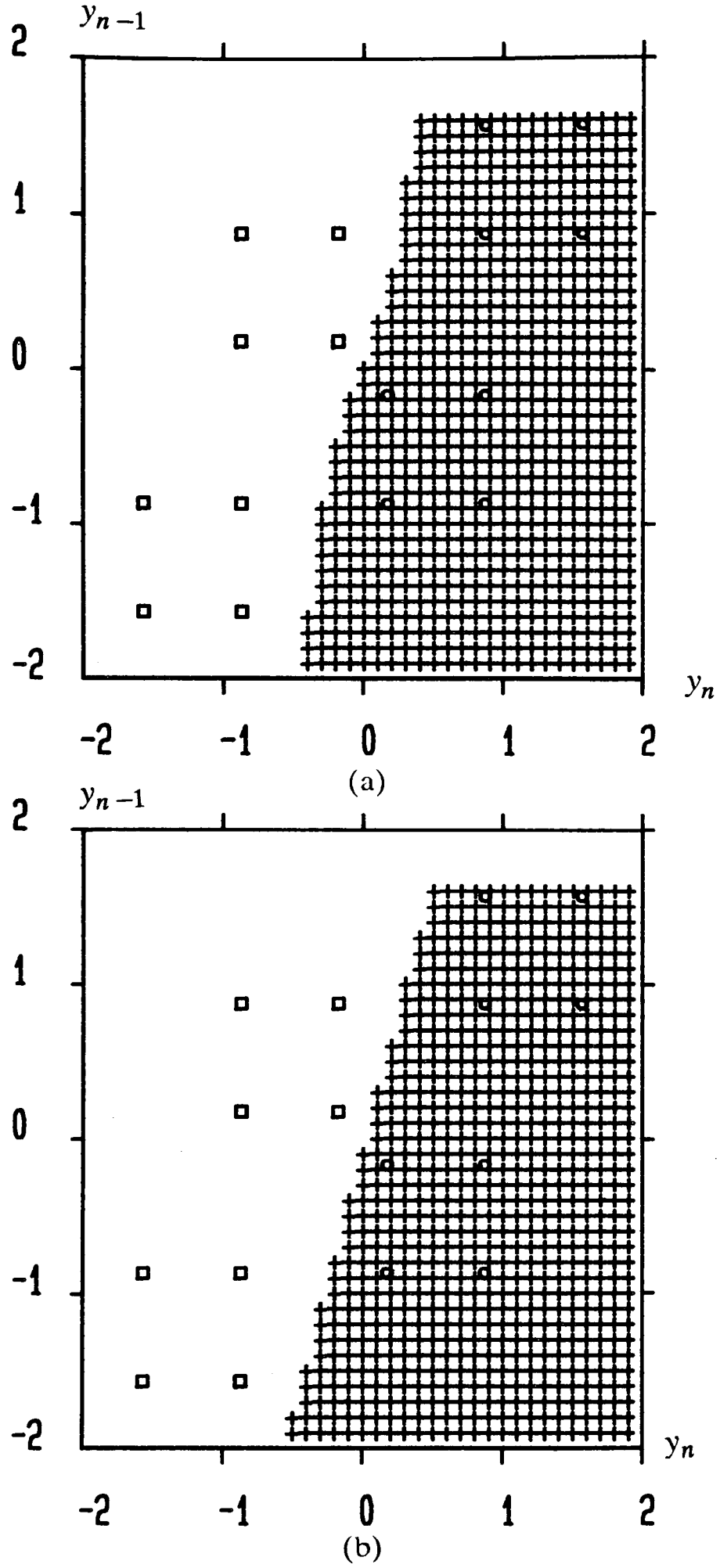


Figure (4.3) Decision regions formed by perceptron-based equalizer { (9,3,1)MLP structure with $N=2$ } after a training period of 1000 samples duration: $\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$.

(a) SNR = 10 dB

(b) SNR = 20 dB

o : signal vector belonging to " 1 ",

□ : signal vector belonging " -1 ".

Shaded area denotes decision region for " 1 ".

sion boundary, rendering it incapable of forming the optimal decision boundary as it does in the low SNR situation.

Examination of Figure (4.4) shows that when the noise level is -20 dB, the decision boundary of the perceptron-based equalizer has a piecewise linear boundary which coincides with the optimal linear equalizer decision boundary " hyperplane" in the central region. Thus, we can expect the performance of the perceptron-based equalizer to be near that of the optimal linear equalizer at high SNR, and to approach it asymptotically in the limit.

4.3.2 Convergence Characteristics

Figure (4.5) illustrates MSE (mean square error) convergence of the perceptron-based equalizer { (9,3,1)MLP structure with order 5 } with $\eta = 0.1$, $\alpha = 0.0, 0.6$, and the LMS linear equalizer (order 5) with $\mu = 0.035$. Here the perceptron with $\eta = 0.1$, $\alpha = 0.0$, converges to noise floor, 20 dB, in about 800 iterations, with $\eta = 0.1$, $\alpha = 0.6$, the convergence time is reduced to about 333 iterations. The results also show that the steady-state value of averaged squared error produced by the perceptron converge to a value lower than the additive noise. This is due to the non-linear nature of the perceptron activation function. The sigmoid function performs a soft decision. It limits the dynamic range of the output signal within [-1,1](see equation (3.6(a))), and also the noise level. Finally, we see the LMS linear equalizer gives a steady-state value of averaged squared error at about 11 dB, above noise floor in 250 iterations. This is due to the fact that the observation vector length in use is insufficient for the equalizer to achieve the noise floor.

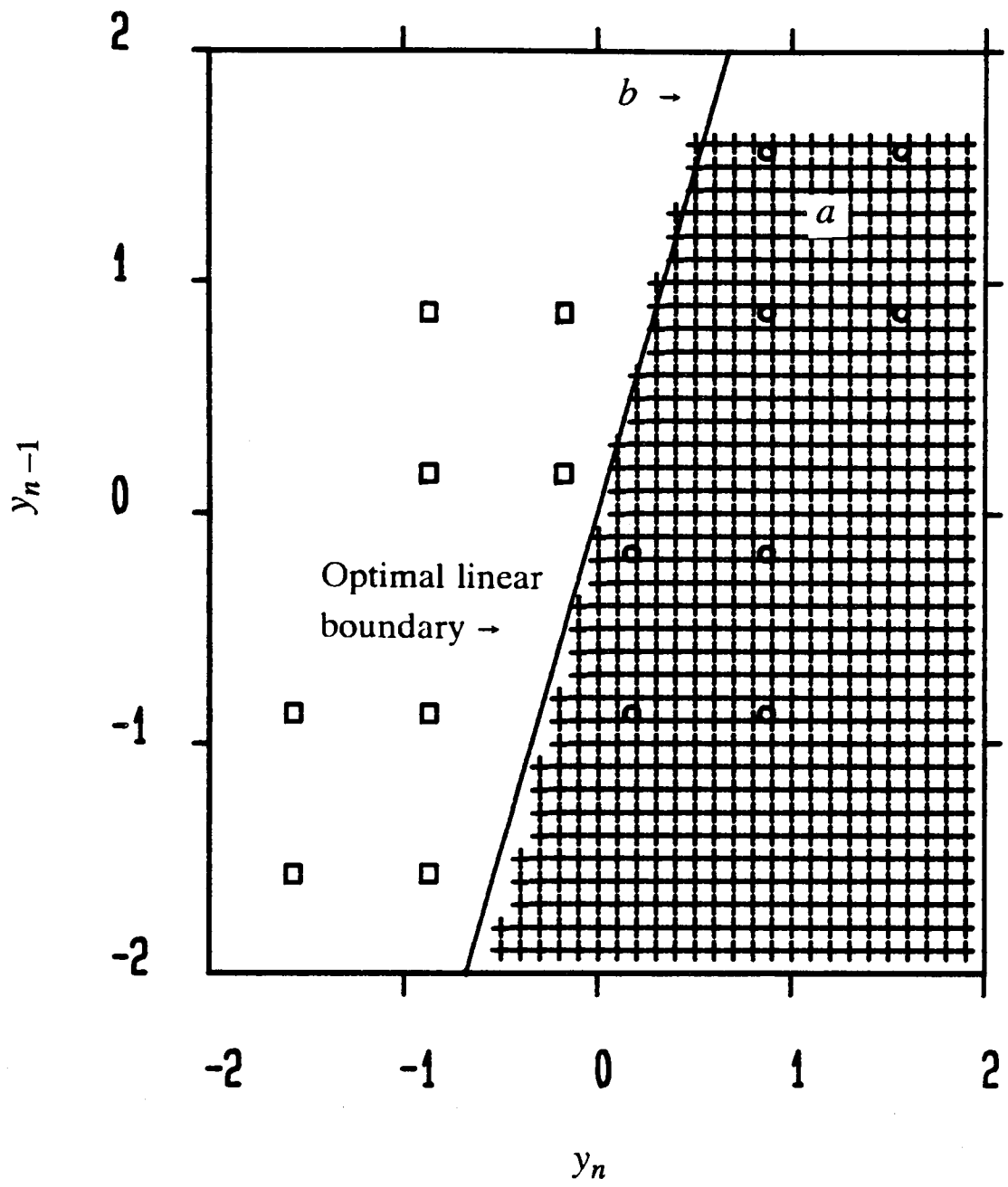


Figure (4.4) Comparison of decision regions formed for SNR=20 dB by

(a) perceptron-based equalizer { (9,3,1)MLP structure with $N=2$ } after a training period of 1000 samples duration : $\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$. Shaded area denotes decision region for " 1 ".

(b) Optimal linear decision boundary (Hyperplane).

o : signal vector belonging to " 1 ",

□ : signal vector belonging to " -1 ".

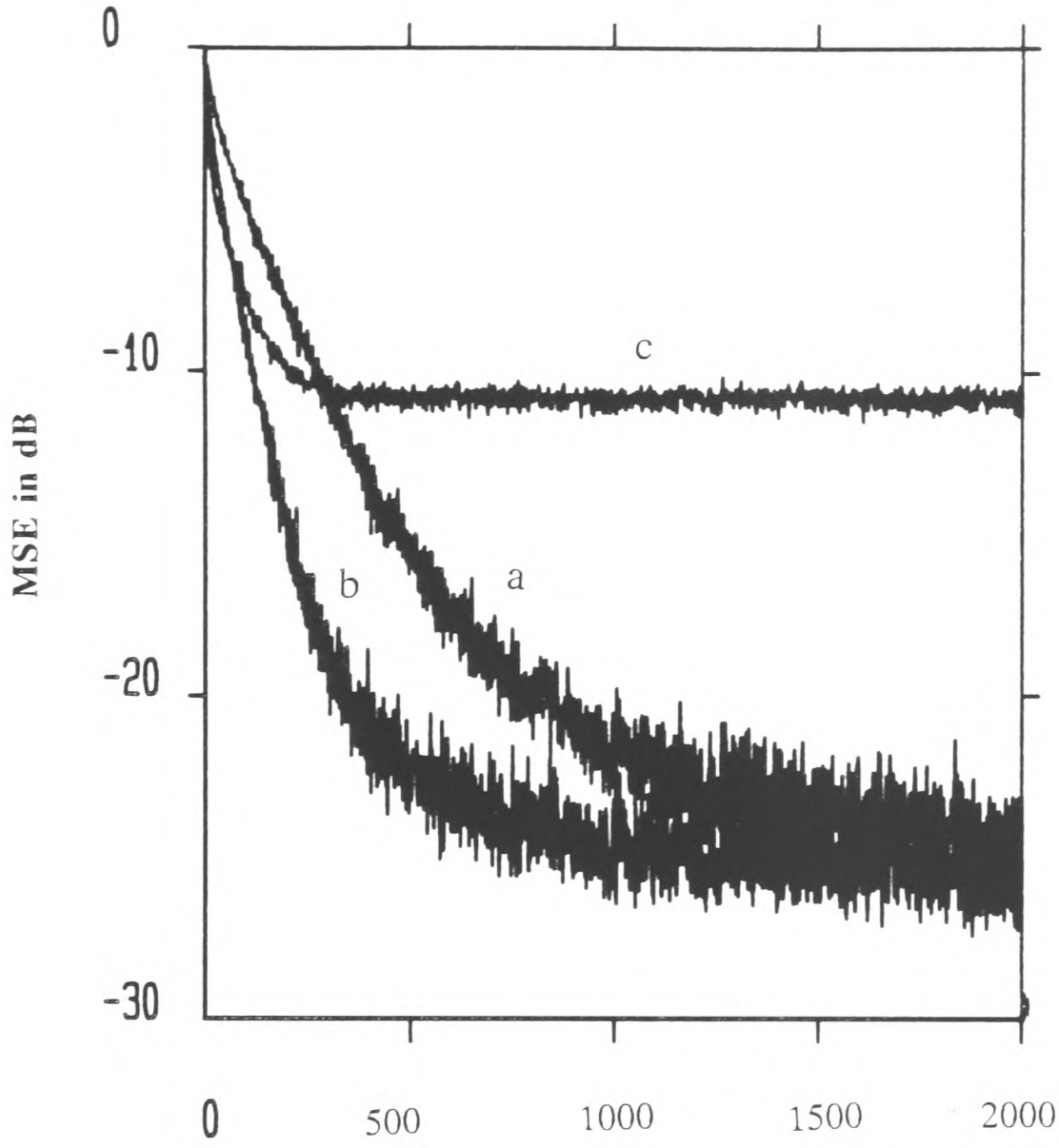


Figure (4.5) Simulation results showing relative convergence rate performance for SNR = 20 dB

(a) { (9,3,1)MLP structure with $N=5$ } : $\eta = 0.1$, $\alpha = 0.0$, $\beta = 0.05$

(b) { (9,3,1)MLP structure with $N=5$ } : $\eta = 0.1$, $\alpha = 0.6$, $\beta = 0.05$

(c){ LMS linear equalizer structure with $N=5$ } : $\mu = 0.035$

4.3.3 Bit Error Rate Performance

4.3.3.1 Ideal Reference mode

In the following simulations comparisons are made of the performance of a perceptron-based equalizer, { (9,3,1)MLP structure with order 2 }, with that of the optimal equalizer by comparing the BER (bit error rate) achieved by the respective structures, over a wide range of signal to noise ratios. The perceptron operates in ideal reference mode, where the data symbols used for adaptation are the correct ones. In order to facilitate a later comparison with the linear transversal equalizer (LTE), a delay of one sample interval is introduced in the estimate of u_i (so that at the i^{th} iteration the equalizer reconstructs an estimate of u_{i-1}). The results of these simulations are shown in Figure (4.6) and they show clearly that the perceptron-based equalizer enjoys a performance which is close to that achieved by the optimal equalizer, when the level of additive noise is high, but deteriorates in comparison as the signal to noise ratio improves.

Further, we compare the performance of the perceptron-based equalizer with the optimal linear error bound and results are presented in Figure (4.7). Examination of Figure (4.7) shows that when the signal to noise ratio is low, the perceptron-based equalizer exhibits better bit error rate performance and it approaches to the the optimal linear error bound as the signal to noise ratio improves. The dynamic behaviour of the decision boundary of the perceptron-based equalizer in Figures (4.2),(4.3.) and (4.4) is futher illustrated by the error rate plots in the results in Figures (4.6) and (4.7).

The final simulations to be presented show clearly the superior performance enjoyed

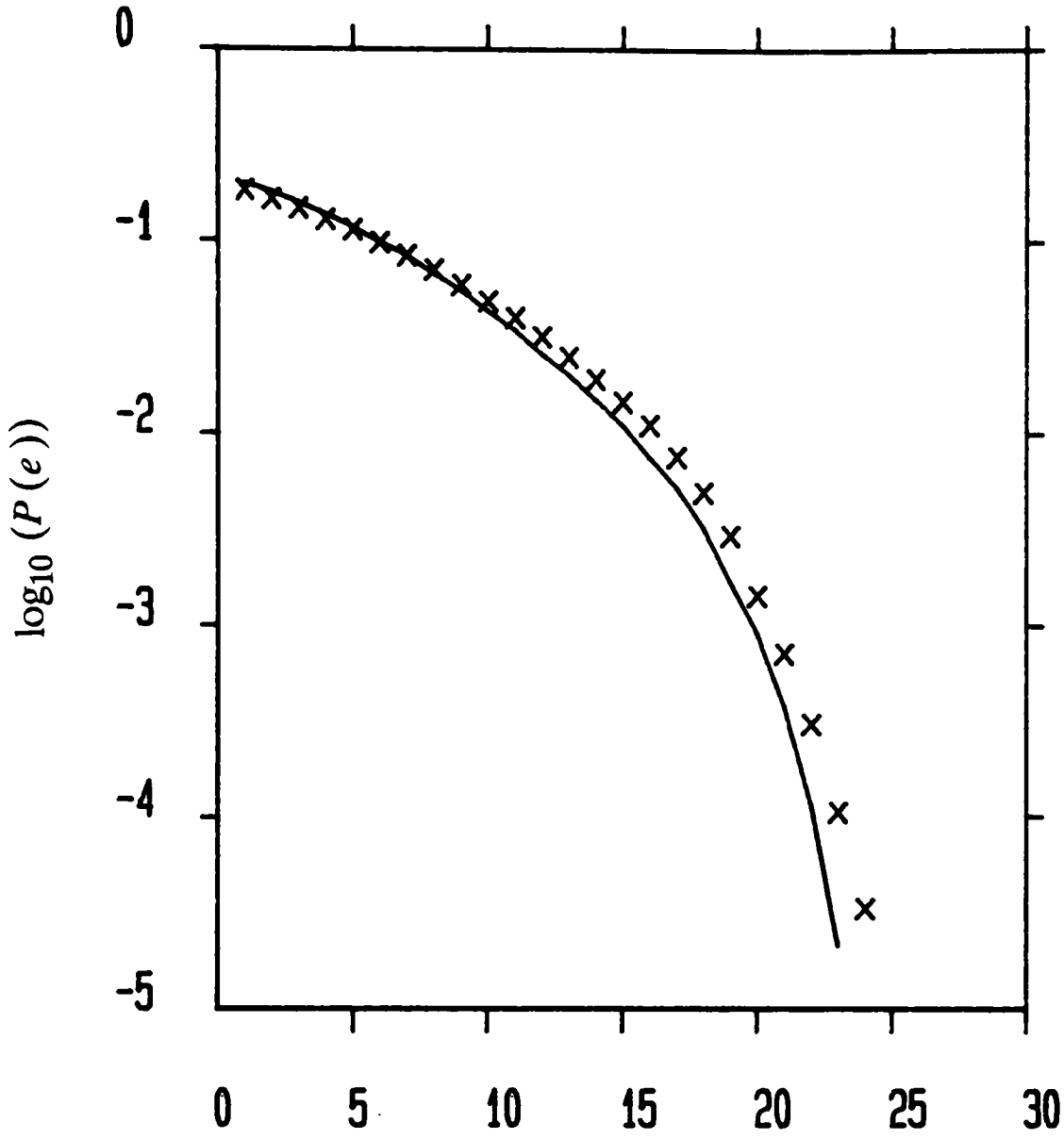


Figure (4.6) Comparison of bit error rate performance achieved by
× : Perceptron- based equalizer {(9,3,1)MLP structure with $N=2$} :
$\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$; ideal reference mode.
– : Optimal equalizer (MAP criterion) with order 2.

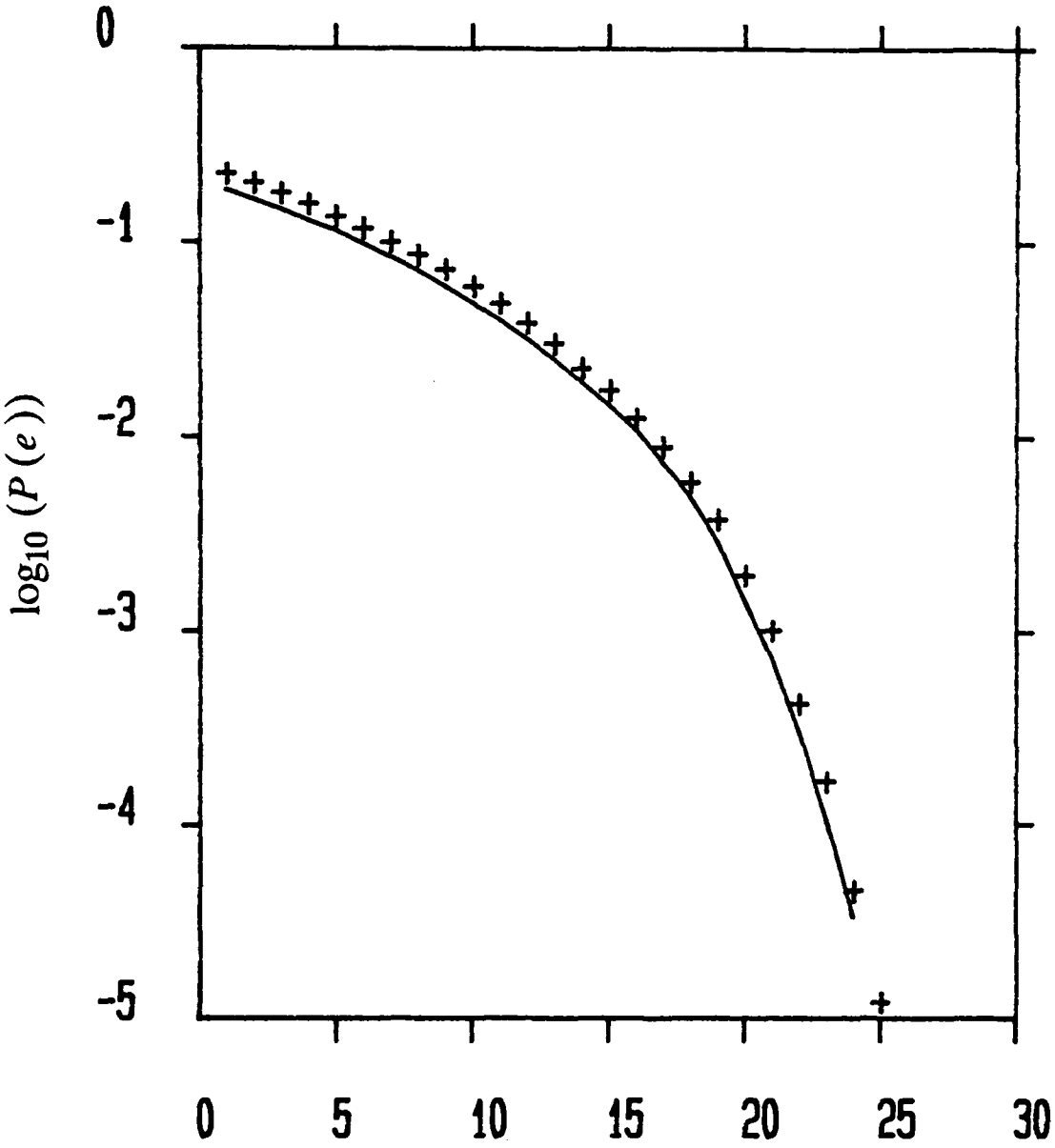


Figure (4.7) Comparison of bit error rate performance achieved by
– : Perceptron-based equalizer { (9,3,1)MLP structure with $N=2$ } :
$\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$; ideal reference mode.
+ : Optimal linear error bound with order 2.

by the perceptron-based equalizer in comparison with the LMS linear equalizer. Figure (4.8) compares the BER achieved by the perceptron-based equalizer, {(9,3,1)MLP structure with order 5 } and the LMS linear equalizer (order 5).

4.3.3.2 Decision-Directed Mode

Perceptron-based equalizer performance can be obtained by means of a Monte Carlo simulation. Through simulations, we have observed the bit error rate performance of the perceptron-based equalizer, { (9,3,1)MLP structure with order 5 }, as a function of the signal to noise ratio for different values of learning parameters (η, α). The perceptron operates in decision-directed mode when the data symbols used for adaptation are the detected symbols.

Figure (4.9) shows the performance of the perceptron with (a) η = 0.1, α = 0.0, 0.5, and β = 0.05, and (b) η = 0.1, 0.2, α = 0.0, and β = 0.05. It may be observed from Figure (4.9) that the choice of parameters in the learning algorithm can affect the bit error rate performance considerably. For the same learning gain, η the bit error rate performance degraded significantly as the signal to noise ratio is decreased for those with a non-zero momentum term (α), as shown in Figure (4.9(a)). However in the case of high signal to noise ratio the performance is slightly improved with a non-zero momentum term (α). In Figure (4.9(b)), shows smaller η provides better bit error rate performance. From the simulation results, it has been concluded that small value for both η and α can provide better bit error rate performance.

Figure (4.10) shows the bit error rate performance curves for both perceptron {(9,3,1)MLP structure with order 5} and LMS linear equalizer (order 5) as a function

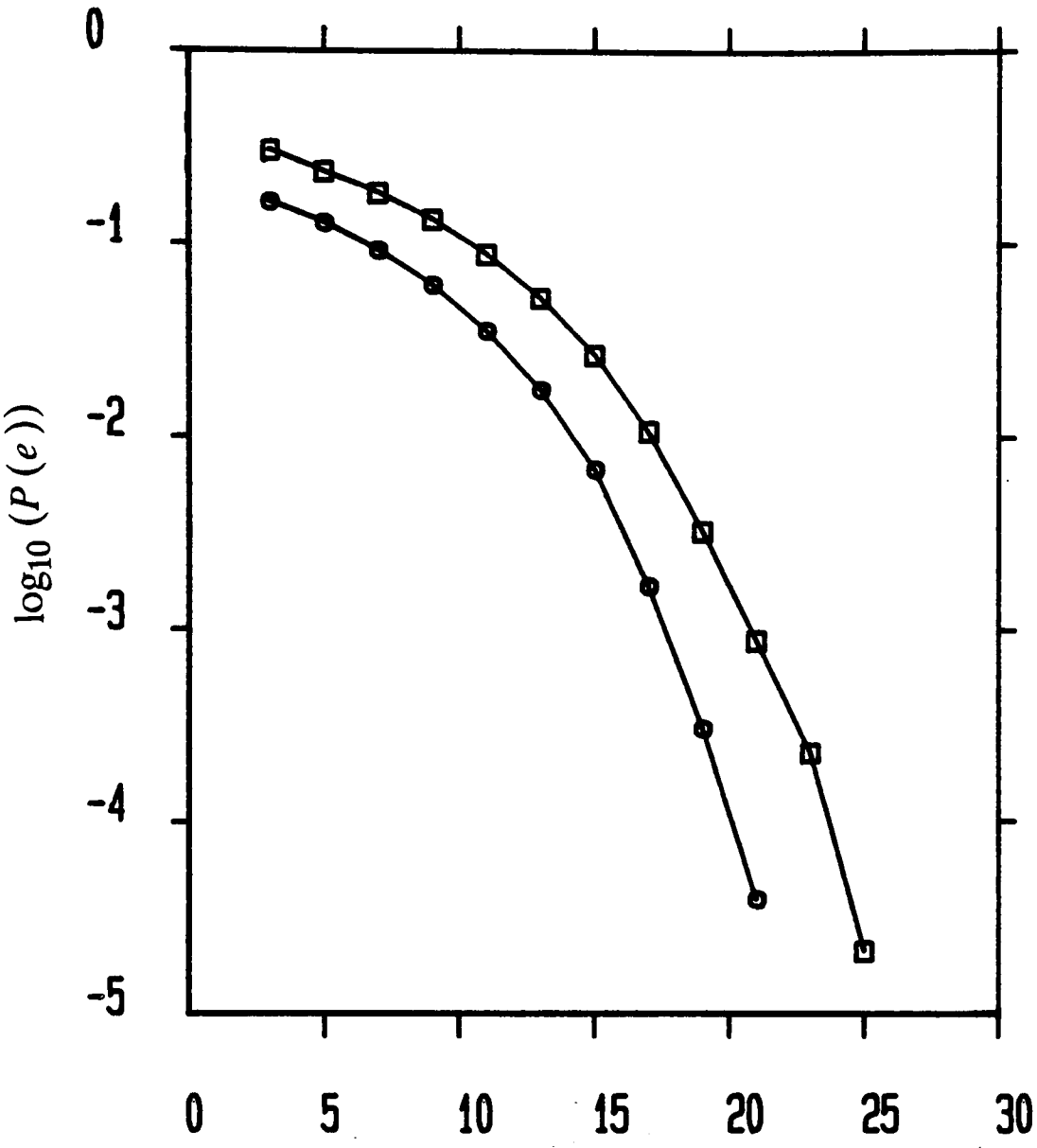
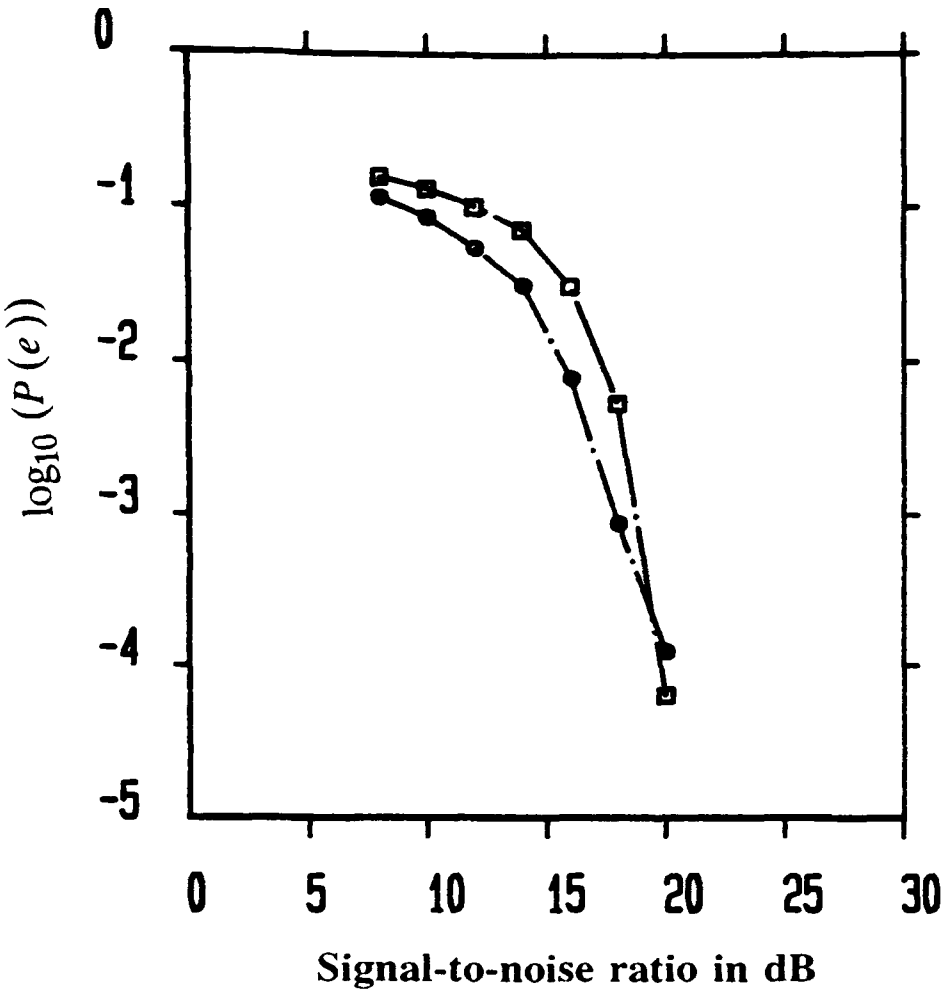


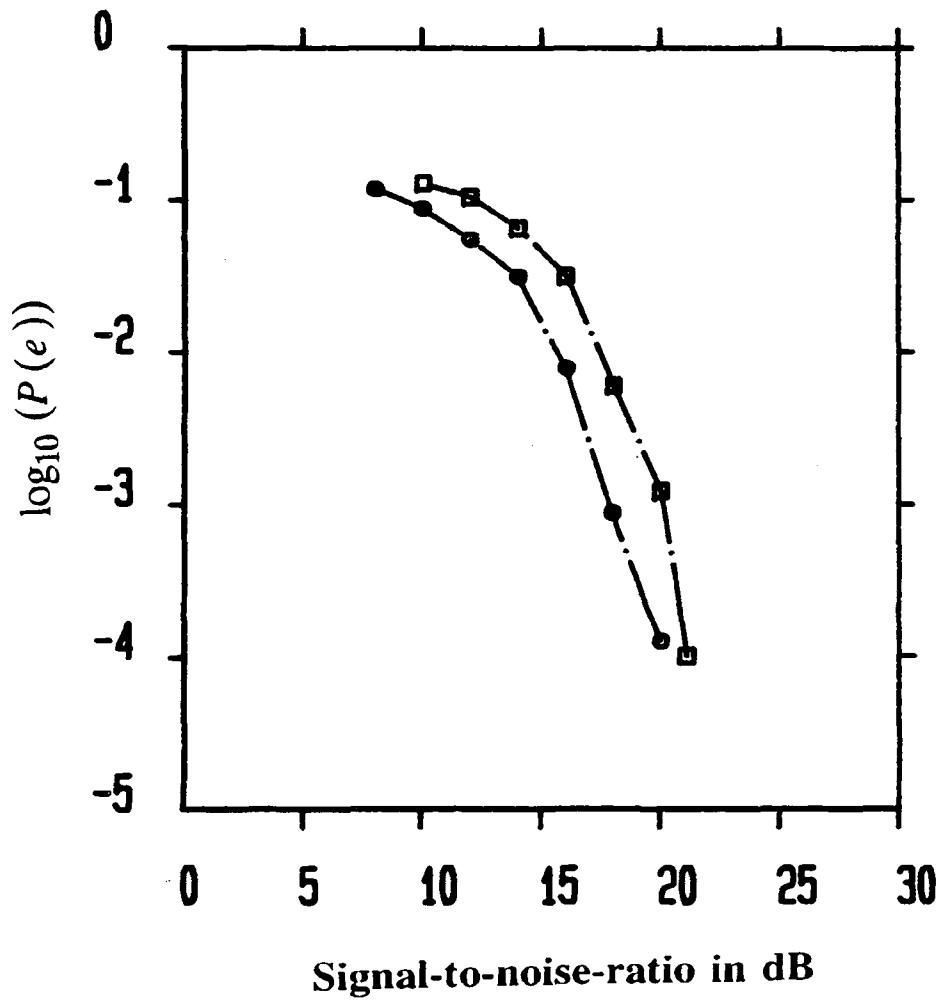
Figure (4.8) Comparison of bit error rate performance achieved by

o : Perceptron-based equalizer { (9,3,1)MLP structure with $N = 5$ } : $\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$; ideal reference mode.

□ : { LMS linear equalizer structure with $N = 5$ } : $\mu = 0.05$; ideal reference mode.



(a)



(b)

Figure (4.9) Bit error rate Performance of perceptron-based equalizer, { (9,3,1)MLP structure with $N=5$ }:

(a) **o** : $\eta = 0.1$, $\alpha = 0.0$, $\beta = 0.05$;

□ : $\eta = 0.1$, $\alpha = 0.5$, $\beta = 0.05$;

(b) **o** : $\eta = 0.1$, $\alpha = 0.0$, $\beta = 0.05$;

□ : $\eta = 0.2$, $\alpha = 0.0$, $\beta = 0.05$.

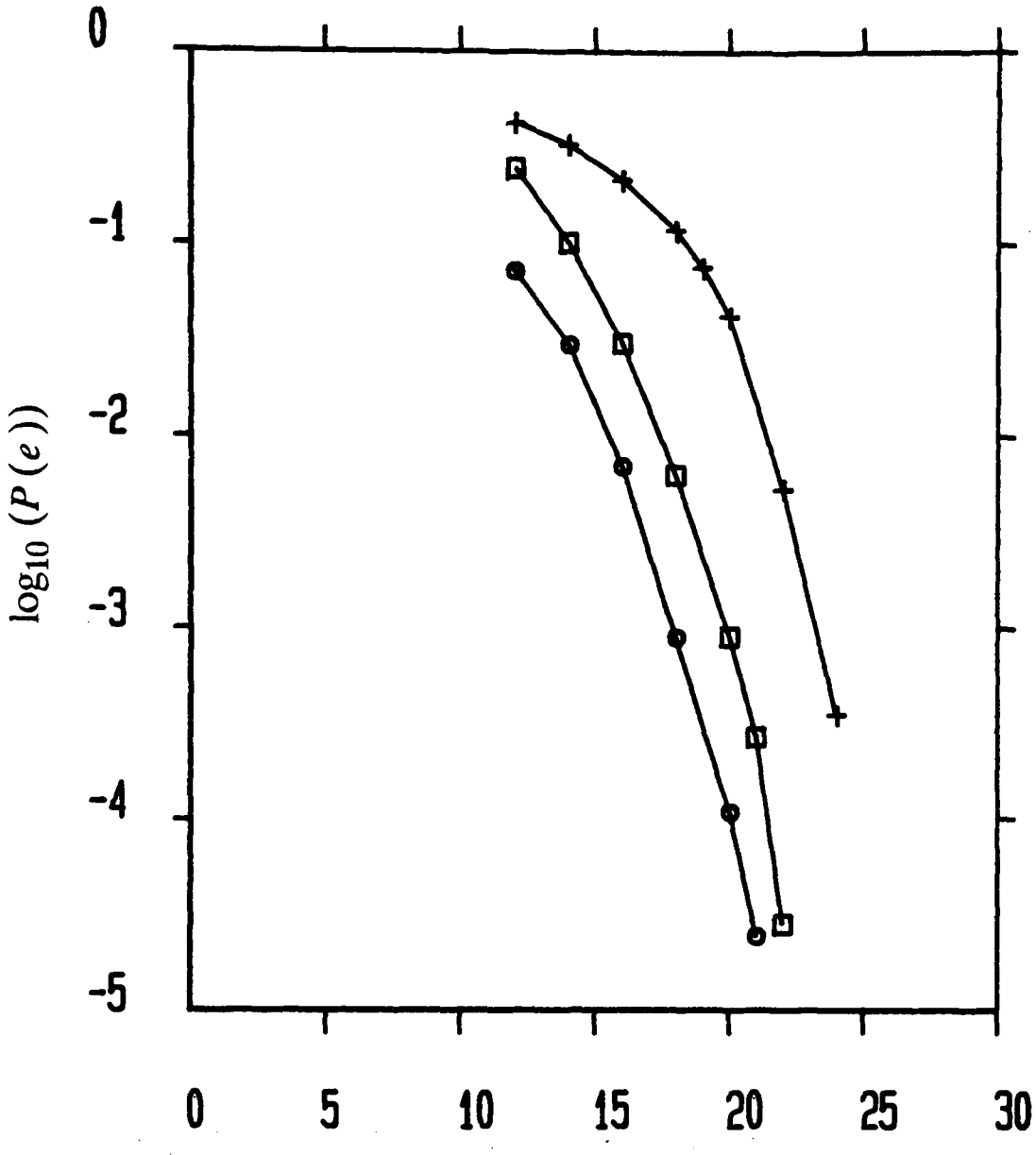


Figure (4.10) Simulation results showing relative bit error rate performance for

Perceptron-based equalizer { (9,3,1)MLP structure with $N = 5$ } :

o : $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$.

{ LMS linear equalizer structure with $N = 5$ }:

+ : $\mu = 0.05$,

□ : $\mu = 0.035$.

of SNR, where $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$, and $\mu = 0.05, 0.035$. The results indicate that the perceptron-based equalizer exhibits better performance. The reason is that the perceptron-based equalizer can achieve a more sophisticated decision boundary (highly non-linear) than a hyperplane. Furthermore, the result also shows that the perceptron is less susceptible to the effects of high levels of additive noise in comparison with the LMS linear equalizer.

4.3.4 Performance Comparison with LMS Linear Transversal Equalizer

The misadjustment of LMS algorithm is defined as [17]

$$M \approx \mu \, tr[\bar{R}] \tag{4.15}$$

Where $\bar{R}$ is the input auto-correlation matrix and tr is the trace operation.

The misadjustment, M is directly proportional to the learning gain constant μ. Thus, there is a trade-off between the misadjustment and the rate of adaptation. Using a small value of μ $(0<\mu<<\frac{1}{tr[R]})$, the LMS process may converge to the steady-state weight solution vector near its optimum $w^*(n)$, then the bit error rate (BER) performance of LMS linear equalizer will be close to the theoretical bound. Thus, we can expect the performance of LMS equalizer to be near that of the perceptron-based equalizer at high SNR(see Figure (4.4)). Figure (4.11) shows the bit error rate performance curves for both perceptron, { (9,3,1)MLP structure with order 5 }, and LMS linear equalizer as a function of SNR, where both values of η and μ are small. Examination of Figure (4.11) shows that the perceptron-based equalizer performs better when the signal to noise ratio is low, and the performance approaches to the LMS linear equalizer while the signal to noise ratio increases. The results in both Figure (4.10) and Figure (4.11) illustrate the above analysis.

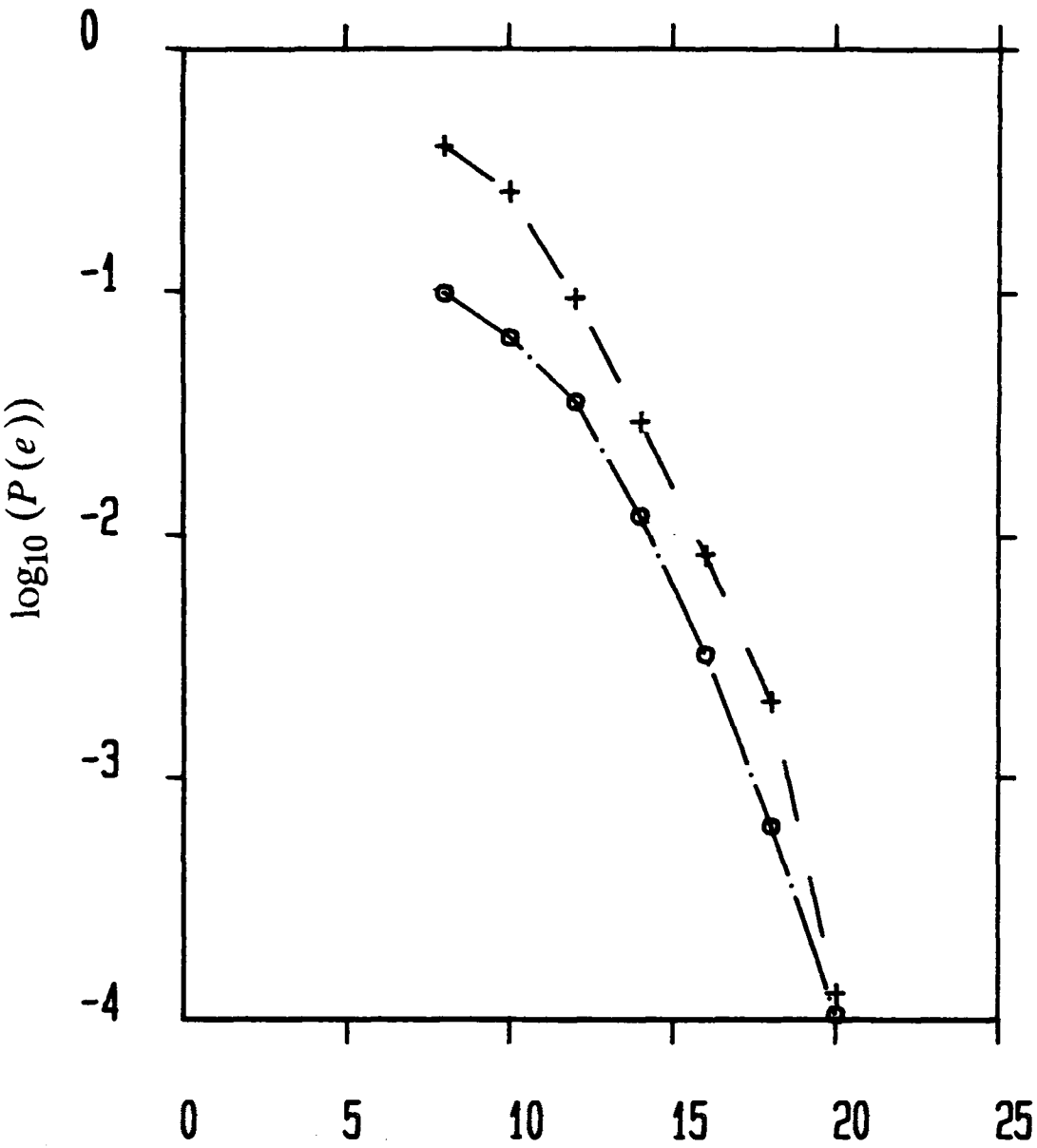


Figure (4.11) Simulation results showing relative bit error rate performance for

Perceptron-based equalizer { (9,3,1)MLP structure with $N = 5$ }:

o : $\eta = 0.05$, $\alpha = 0.2$, and $\beta = 0.05$,

{ LMS linear equalizer structure with $N = 5$ }:

+ : $\mu = 0.025$.

4.4 Effect of Network Topology on the Performance of the System

In this section, we explore the effect of network topology (e.g., input dimension, number of neurons, number of hidden layers) on the performance (e.g., convergence time, bit error rate) of a multi-layer perceptron-based equalizer.

4.4.1 Input Dimension and Number of Neurons

At present no theory or method exists to predict the optimal configuration. Most of the results are obtained by simulations. A series of simulations were performed to study the effect of the input dimension (N) and the number of neurons (N_1 and N_2) within each hidden layer on the performance of the perceptron-based equalizer { (N_1,N_2,N_3) MLP structure with order N }, in SNR = 20 dB, η = 0.07, α = 0.3, and β = 0.05.

Figure (4.12) shows the bit error rate as a function of the input dimension with (9,3,1)MLP structure. It can be seen that increasing the order N from 3 to 5 has the effect of reducing the BER performance and the BER curve shows no significant change after N = 5.

Figures (4.13) and (4.14) show the effect of extra neurons within each hidden layer on the error rate and the convergence time for both (N_1,3,1)MLP, and (9,N_2,1)MLP structures with order 5. The results show that for each of the structures shown, as the number of the neurons within each each hidden layer is increased, there is initially a very significant improvement in learning speed, but this improvement becomes less significant as the number of neurons further increases. It can be seen that the convergence time for learning in three-layer perceptron is faster than the in two-layer perceptron. The bit error rate performance curves show little change as the number

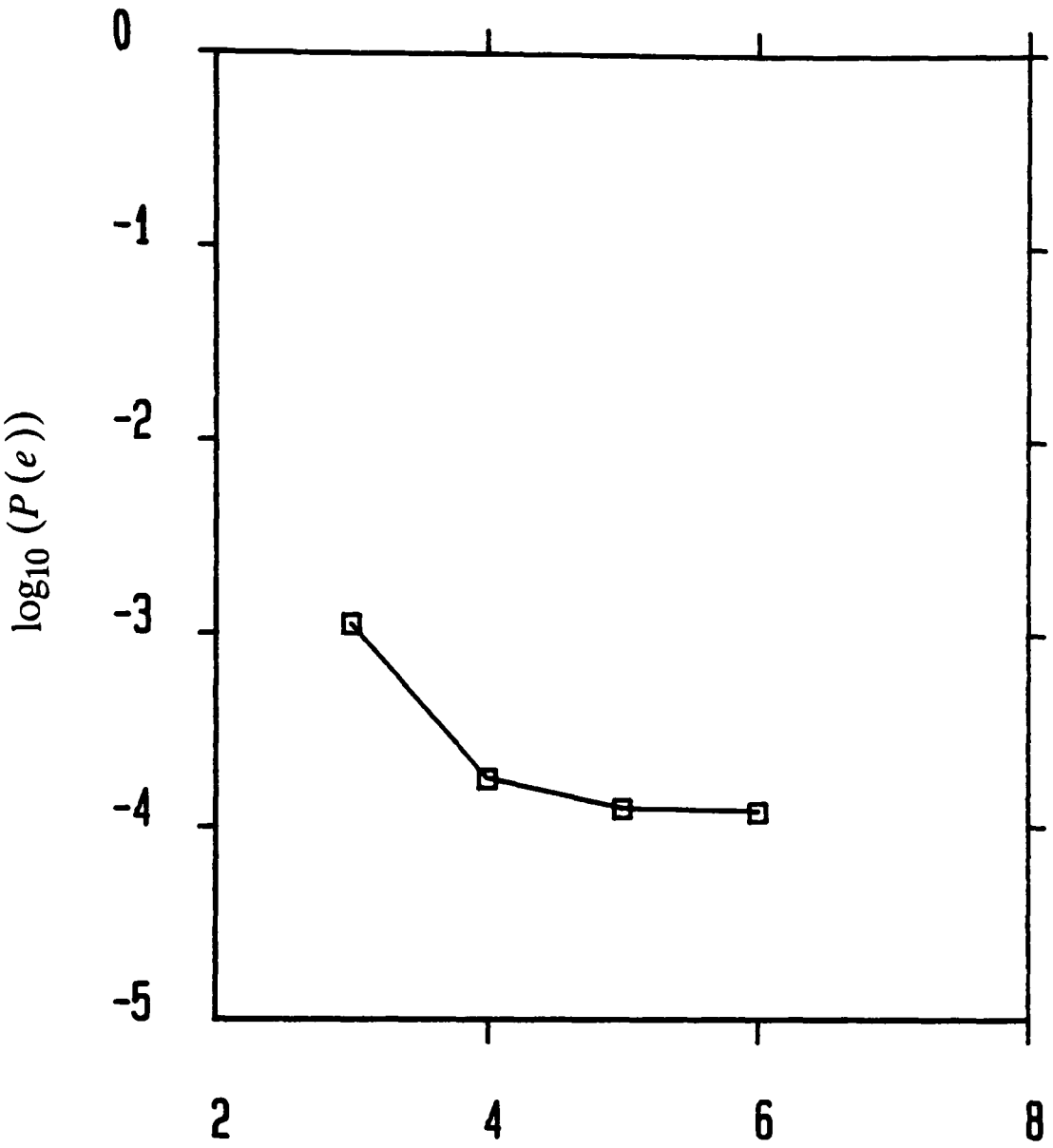
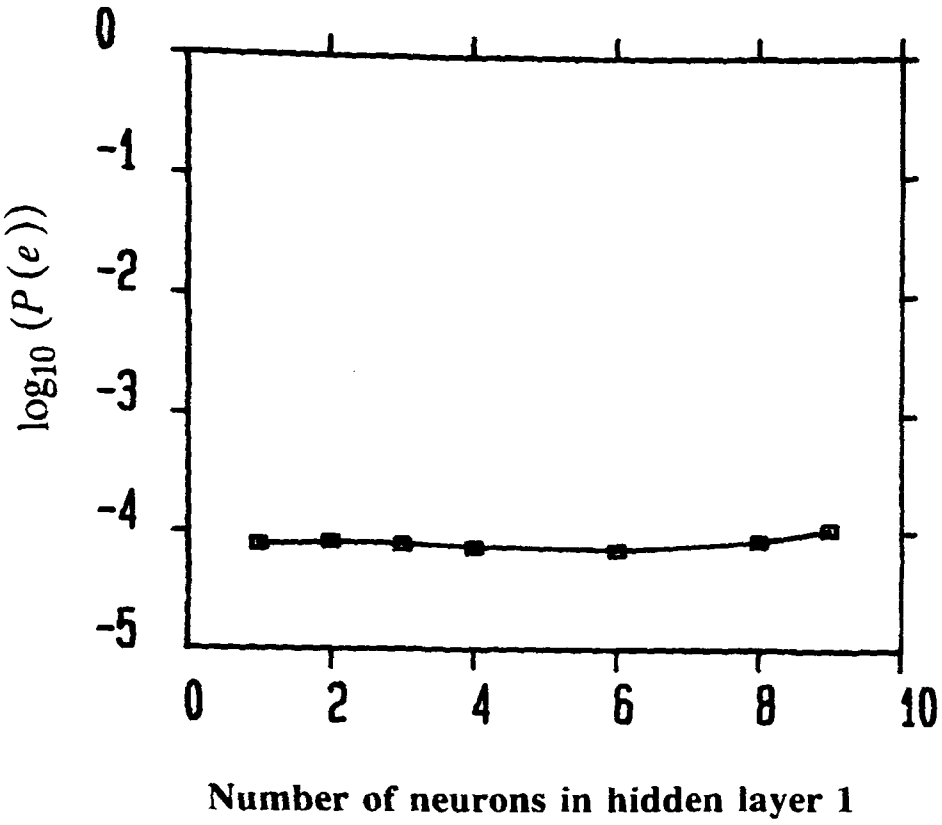
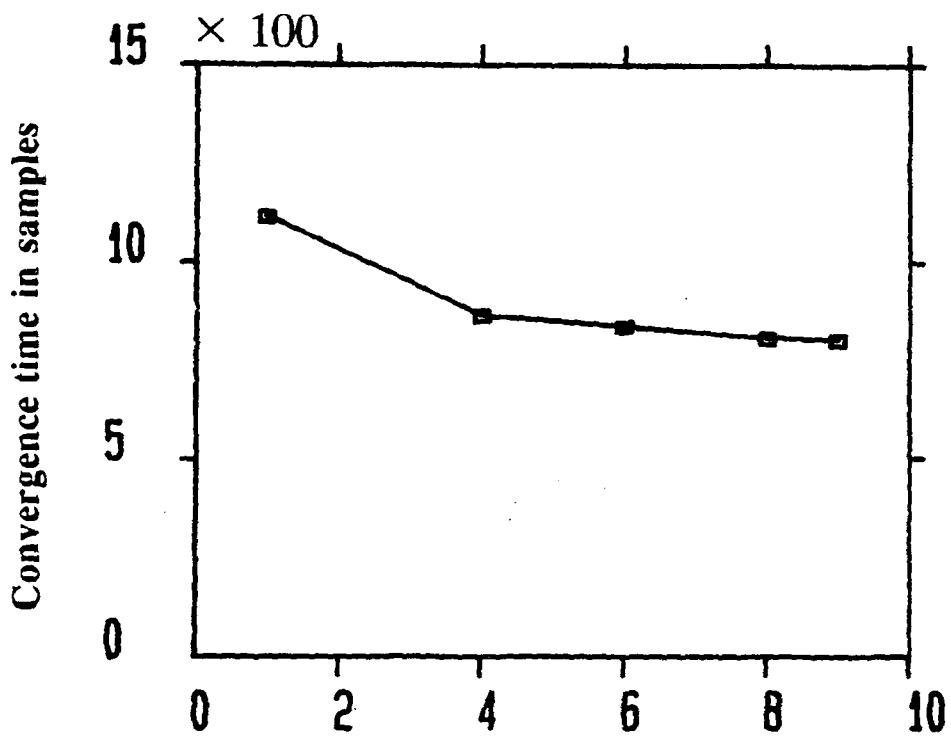


Figure (4.12) Bit error rate performance as a function of input dimension (N), for perceptron-based equalizer, { (9,3,1)MLP structure with N } at SNR = 20 dB : $\eta = 0.03$, $\alpha = 0.3$, $\beta = 0.05$.



(a)

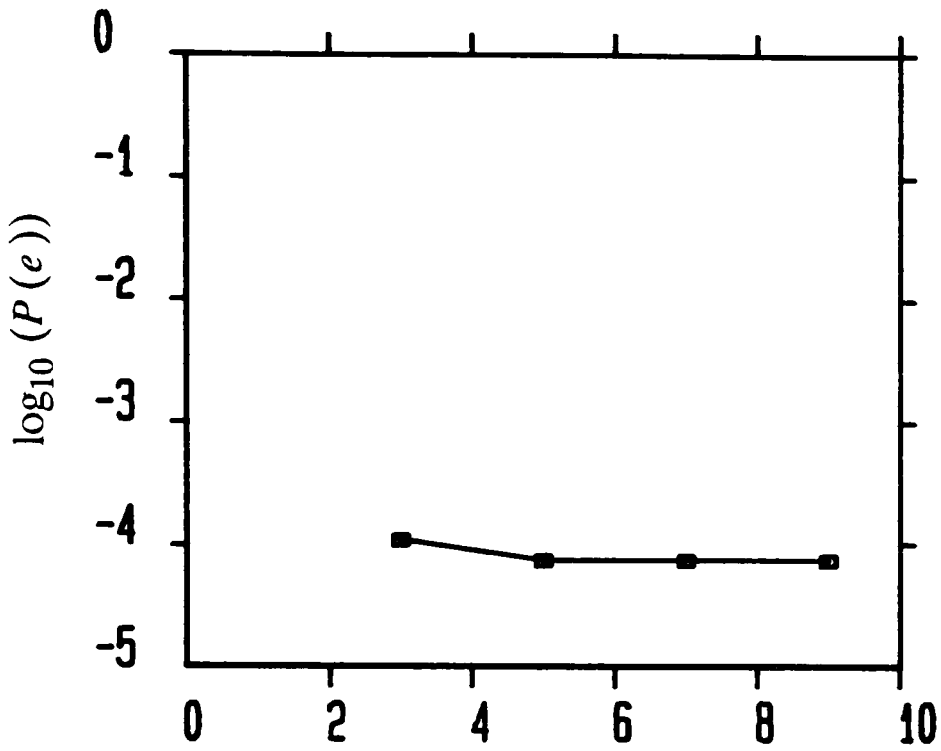


(b)

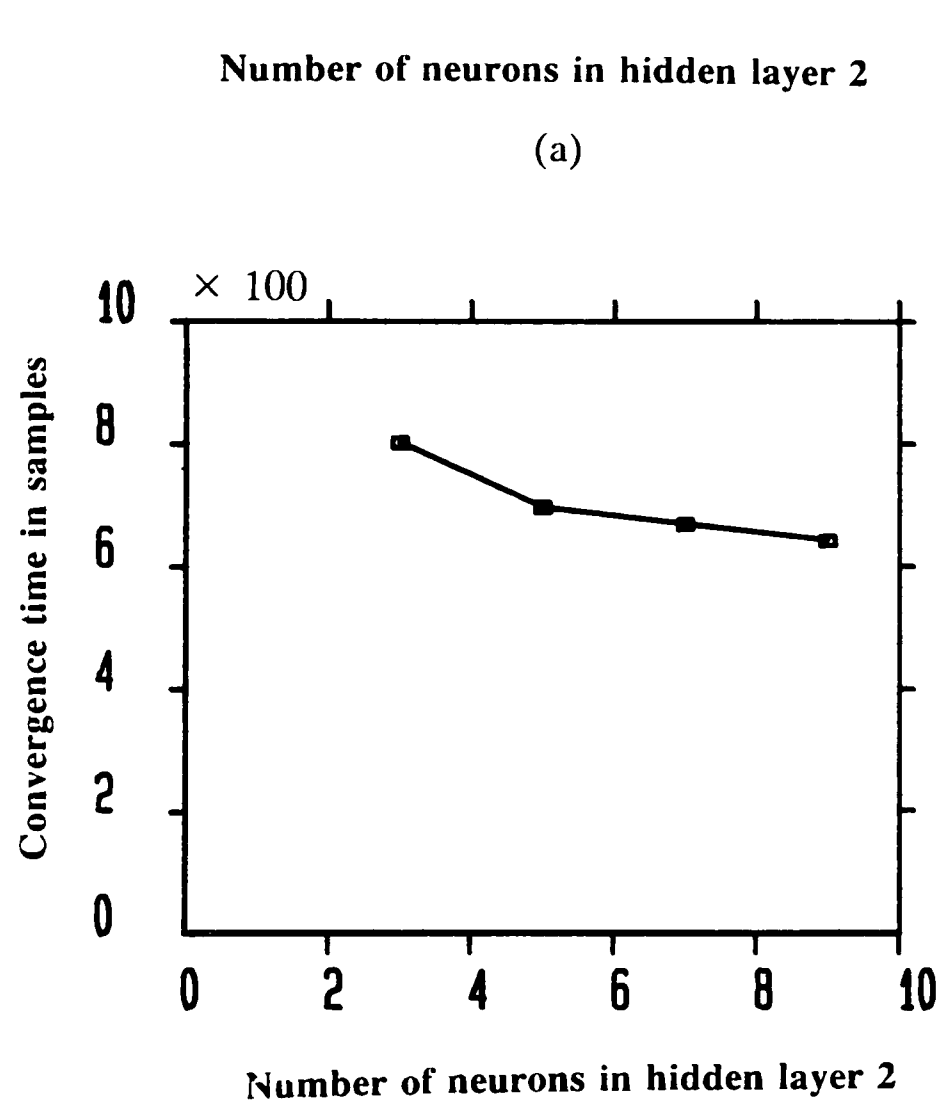
Figure (4.13) Performance as a function of N_1 (number of neurons in the first hidden layer), for perceptron-based equalizer { (N_1 ,3,1) MLP structure with $N=5$ } at SNR = 20 dB

(a) Bit error rate in $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$,

(b) Convergence time to the noise floor.



(a)



(b)

Figure (4.14) Performance of as a function of N_2 (number of neurons in the second hidden layer), for perceptron-based equalizer { (9, N_2 , 1)MLP structure with $N = 5$ } at SNR = 20 dB

(a) Bit error rate in $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$,

(b) Convergence time to the noise floor.

of neurons increase within each hidden layer. Similar results for three-layer perceptron structures were observed in [44],[56],[57].

4.4.2 Comparative results of two-layer and three-layer

In chapter 3, we show that two-layer perceptrons are capable of forming convex, non-convex, and disconnected decision regions. In this section, we compare the bit error rate performance of the three-layer perceptron,{ (9,3,1)MLP structure with order 2 }, and the two-layer perceptron, { (9,1)MLP structure with order 2 } including the optimal linear equalizer. Figure (4.15), shows that the three-layer perceptron gives a slight improvement in bit error rate performance relative to the two-layer perceptron, especially in the low signal to noise ratio environment. The reason is that the former structure with one extra hidden layer can form more non-linear decision boundaries than the latter. Examination of Figure (4.16) shows that when the signal to noise ratio is low, the two-layer perceptron performs slightly better relative to the optimal linear error rate bound, but the improvement reduces as the signal to noise ratio improves. Finally it approaches to the optimal linear error rate bound which is based on optimized the decision boundary (equation (2.25)).

4.5 Discussion

The perceptron-based equalizer offers advantage over the conventional linear equalizer. It has the ability to form nonlinear decision regions , in contrast with the linear equalizer which can only form linear decision regions. The linearity of the decision regions limits the performance of the linear equalizer, especially in poor noise conditions.

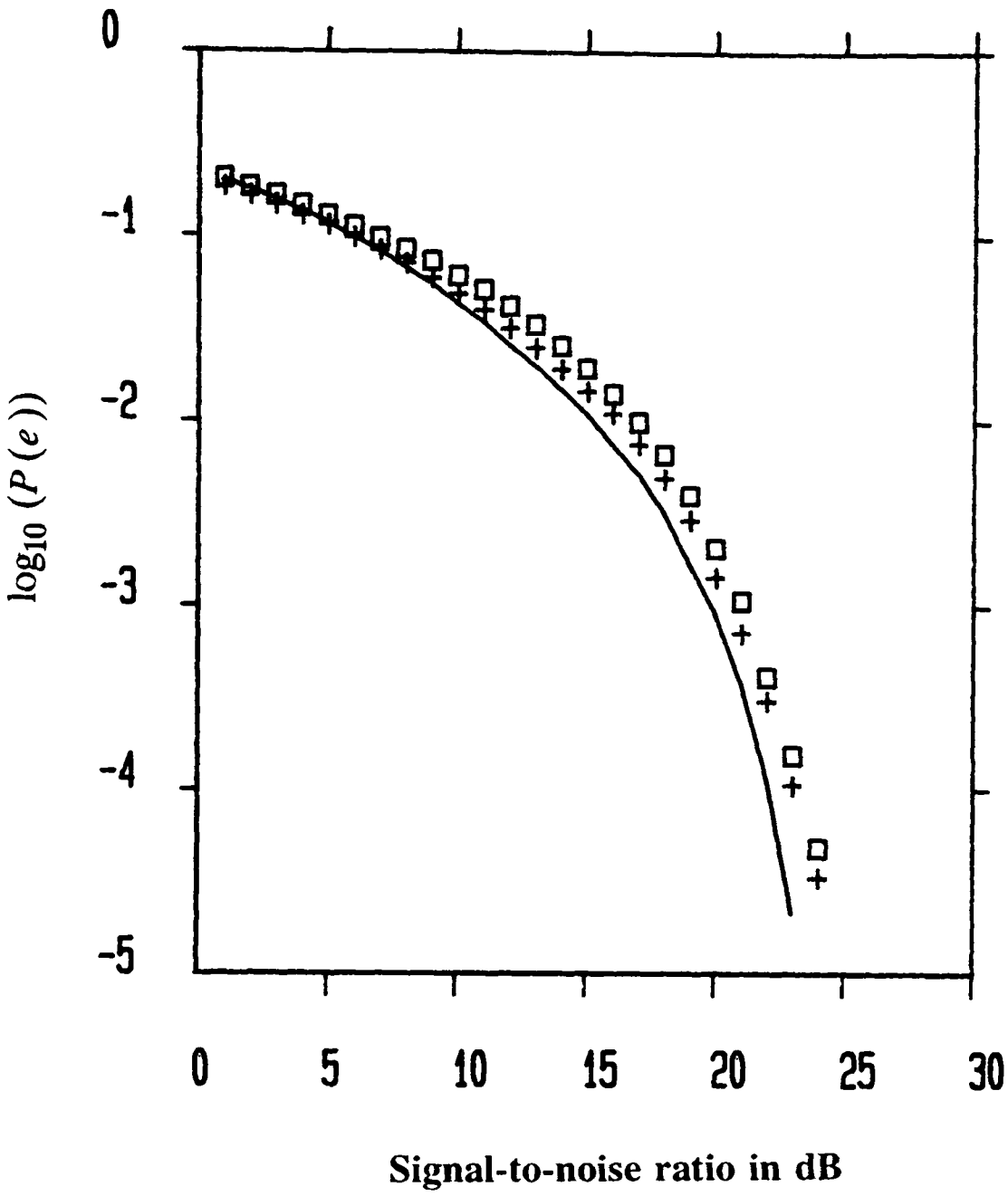


Figure (4.15) Comparison of bit error rate performance achieved by

+ : Perceptron-based equalizer {(9,3,1)MLP structure with $N=2$ }:
$\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$; ideal reference mode.

□ : Perceptron-based equalizer { (9,1)MLP structure with $N=2$ } :
$\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$; ideal reference mode.

– : Optimal equalizer (MAP criterion) with order 2.

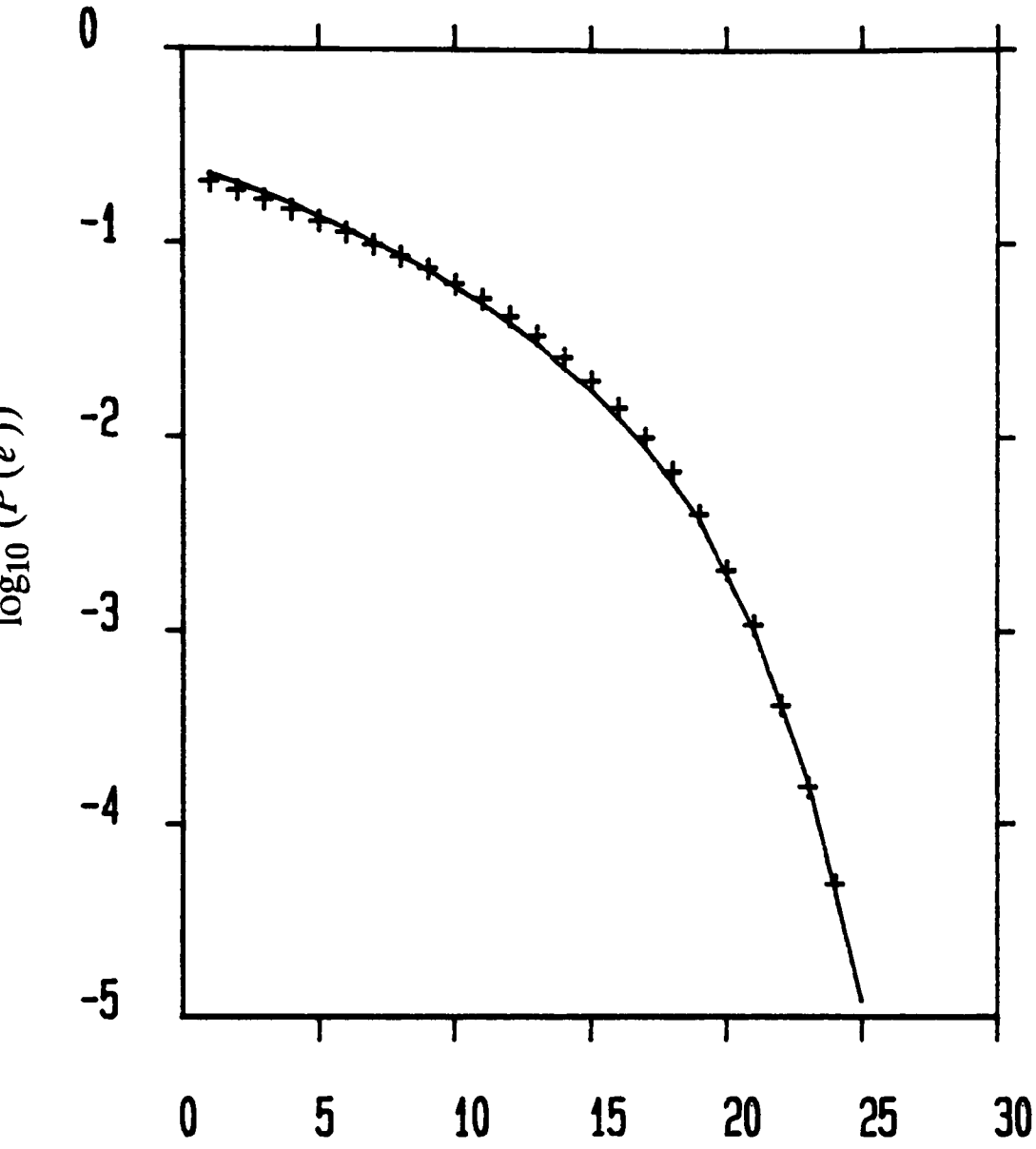


Figure (4.16) Comparison of bit error rate performance achieved by

+ : Perceptron-based equalizer { (9,1)MLP structure with $N = 2$ } :
$\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$; ideal reference mode.

− : Optimal linear error bound with order 2.

The perceptron-based equalizer is less susceptible to the effects of high levels of additive noise relative to the linear equalizer. The simulation results also show the bit error rate performance of perceptron-based equalizer to be near that of LMS linear transversal equalizer at a "high" SNR, when the learning gain (μ) in the LMS algorithm is very small.

Both two-layer and three-layer perceptrons are capable of forming convex, non-convex, and disconnected decision regions. The three-layer perceptron gives a slight improvement in bit error rate relative to the two-layer perceptron, especially in the low signal to noise ratio environment. Furthermore, the convergence time for learning in the three-layer perceptron is faster than in the two-layer perceptron.

For the perceptron-based equalizer, as the number of neurons within each hidden layer is increased, there is initially a very significant improvement in learning speed, but this improvement becomes less significant as the number of neurons further increases. The bit error rate performance curves show little change as the number of neurons increases within each hidden layer.

CHAPTER 5

A PERCEPTRON - BASED

DECISION FEEDBACK EQUALIZER

5.1 Introduction

Decision feedback equalization [1],[58] is a technique used in digital communications systems (see Figure(5.1)) to equalize the channel to remove that part of the intersymbol interference (ISI) caused by the previous data decisions. The advantage of the decision feedback equalizer is that ISI is eliminated without enhancement of noise by using past decisions to subtract out a portion of the ISI; a disadvantage is that decision errors tend to propagate because they result in residual ISI and a reduced margin against noise at future decisions [59],[60],[61].

The conventional structure of the decision feedback equalizer (DFE) uses linear algorithms such as LMS (Least mean squares) or RLS (recursive least squares) and consists of a feedforward filter and a feedback filter, as shown in Figure (5.1), where the feedforward filter is a linear equalizer. The decision regions of a linear equalizer are always delimited by hyperplanes. The linear nature of the decision boundaries limits the performance of the DFE. The multi-layer perceptron may be simply considered as a nonlinear mapping from input to output. The hidden layers provide the capability,

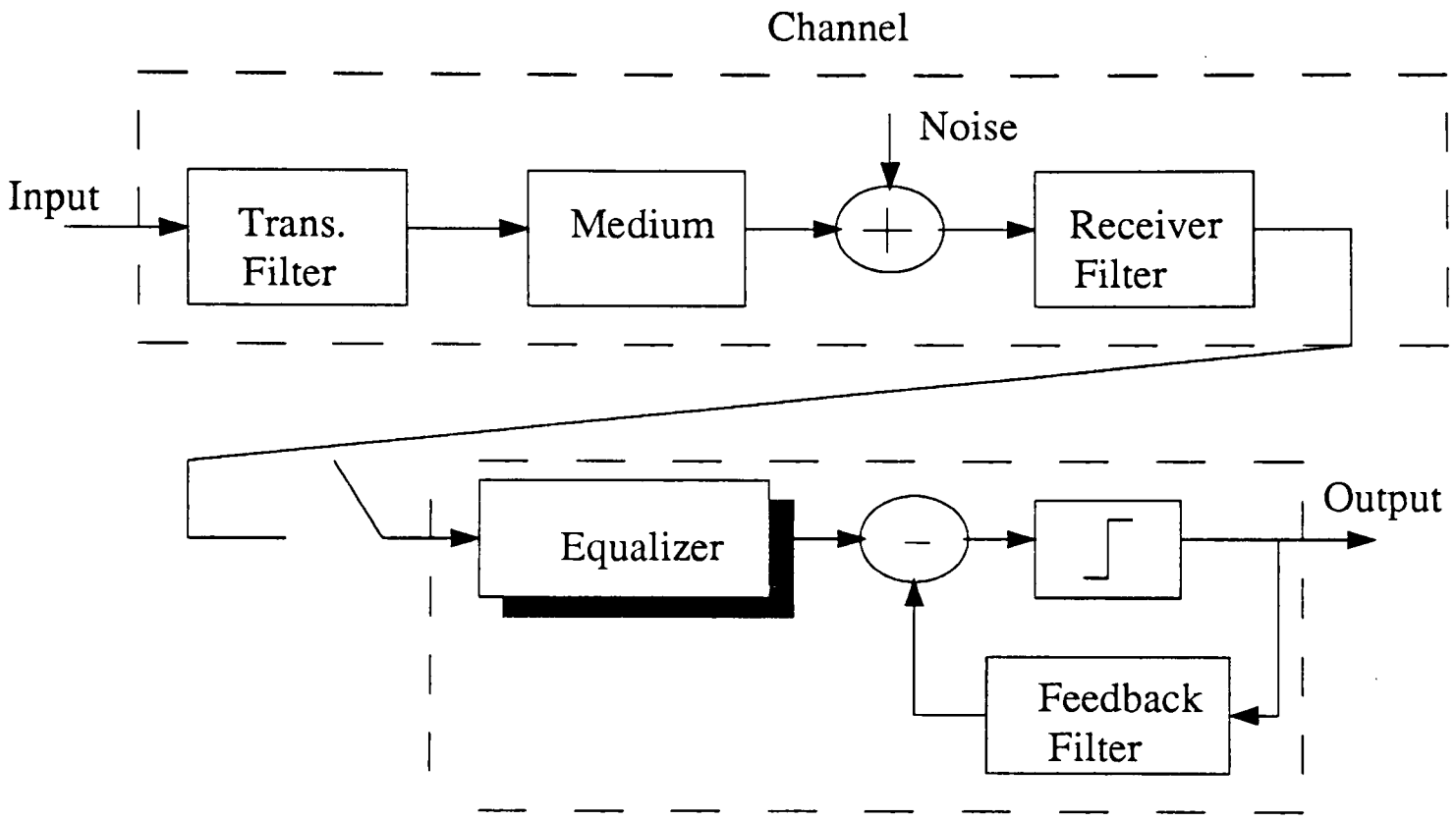


Figure (5.1) A baseband data transmission system with decision feedback equalizer

by use of the nonlinear sigmoid function, to create intricately curved partitioning of the input signal space to produce highly nonlinear decision boundaries (equation (3.5)).

A new architecture for a decision feedback equalizer using the multi-layer perceptron structure for equalization in digital communication systems is presented [62]. The topics addressed are; architecture, modelling, learning algorithm, intersymbol interference cancellation, and performance (convergence characteristics and bit error rate). Comparisons of the performance of this perceptron-based DFE with a conventional DFE (i.e. LMS DFE) are provided. The simulation results demonstrate the superior performance of the multi-layer perceptron based DFE. The improvement is significant, especially in high noise conditions.

5.2 Perceptron-Based DFE

5.2.1 Architecture, and Modelling

A three-layer perceptron based decision feedback equalizer structure, as shown in Figure (5.2), consists of a feedforward filter and a feedback filter. The input to the feedforward filter is the sequence of noisy received signal samples $\{ y_n \}$. The input to the feedback filter is the output symbol decision sequence from a nonlinear symbol detector (quantiser) $\{ \tilde{u}_{n-d} \}$.

At time n, the input $N \times 1$ received signal vector

$$\Gamma(n)^T = [y_n, y_{n-1} \cdots . y_{n-N+1}] \tag{5.1}$$

and the decision $l \times 1$ signal vector

$$[\tilde{u}_{n-d-1}, \tilde{u}_{n-d-2}, \ldots, \tilde{u}_{n-d-l}] \tag{5.2}$$

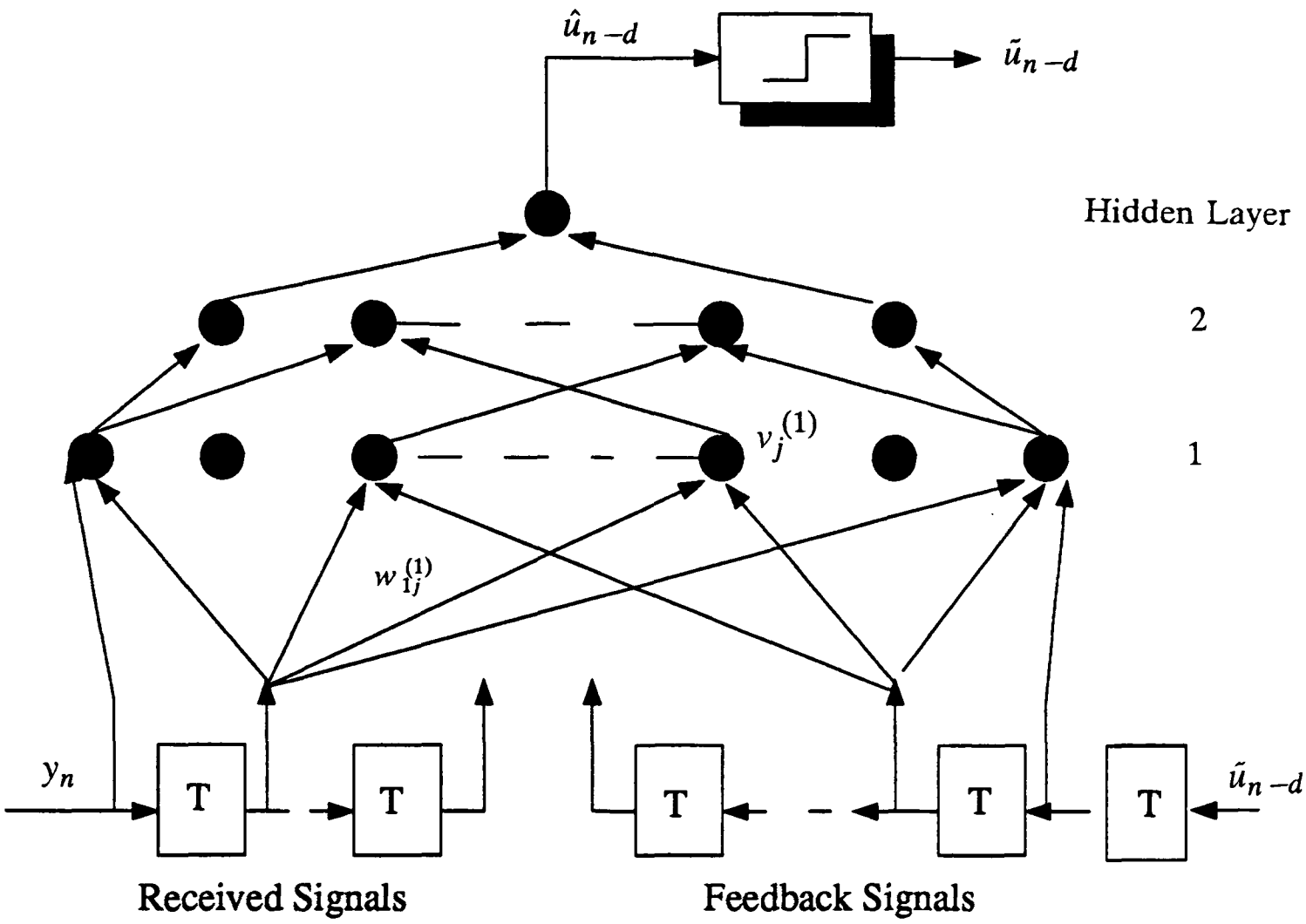


Figure (5.2) Multi-layer perceptron decision feedback equalizer

are in the feedforward filter and feedback filter of the decision feedback equalizer, respectively, where d is a delay parameter. The decision $\tilde{u}_{n-d}$ is formed by quantizing the estimate $\hat{u}_{n-d}$ in the output layer to the nearest information symbol.

The signals at the input layer of the decision feedback equalizer can be represented by a $(N+l)\times 1$ vector as

$$V^{(0)} = \left[y_n \; y_{n-1} \; . \; . \; y_{n-N+1} \; \tilde{u}_{n-d-1} \; . \; . \; \tilde{u}_{n-d-l} \right]^T \tag{5.3}$$

The $N_1 \times 1$ vector in the output of hidden layer 1 is :

$$V^{(1)} = \left[v_1^{(1)} \; v_2^{(1)} \; . \; . \; v_j^{(1)} \; . \; . \; v_{N_1}^{(1)} \right]^T \tag{5.4}$$

where

$$v_j^{(1)} = f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} \, y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)} \, \tilde{u}_{n-d-p} + I_j^{(1)} \right) \qquad j = 1,2,.....N_1 \tag{5.5}$$

Where b denotes the feedback tap weight.

The $N_2 \times 1$ vector in the output of hidden layer 2 is :

$$V^{(2)} = \left[v_1^{(2)} \, , \; v_2^{(2)} \, , \; . \; . \; v_k^{(2)} , \; . \; . \; v_{N_2}^{(2)} \right]^T \tag{5.6}$$

where

$$v_k^{(2)} = f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} \, v_j^{(1)} + I_k^{(2)} \right) \qquad k = 1,2,.....N_2 \tag{5.7}$$

The final output is :

$$v_o^{(3)} = \hat{u}_{n-d} = f_o\left(\sum_{k=1}^{N_2} w_{ko}^{(3)} \, v_k^{(2)} + I_o^{(3)} \right) \tag{5.8}$$

Where $\hat{u}_{n-d}$ is the estimated signal at time n. Substituting equations (5.5) and (5.7)

into equation (5.8), yields

$$\hat{u}_{n-d} = f_o\Big(\sum_{k=1}^{N_2} w_{ko}^{(3)}\, f_k\Big(\sum_{j=1}^{N_1} w_{jk}^{(2)}\, f_j\Big(\sum_{i=0}^{N-1} w_{ij}^{(1)}\, y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)}\, \tilde{u}_{n-d-p} + I_j^{(1)}\Big) + I_k^{(2)}\Big) + I_o^{(3)}\Big) \tag{5.9}$$

The nonlinear detector can be modelled as a threshold function $g(x)$ and is defined as

$$g(\hat{u}_{n-d}) = \tilde{u}_{n-d} = \begin{cases} 1 & \text{if } \hat{u}_{n-d} \geq 0 \\ -1 & otherwise \end{cases} \tag{5.10}$$

The $w's$ (weights) and $I's$ (threshold levels) in equation (5.9) are values specified by the training algorithm, so that after training is finished the equalizer will self-adapt to changes in channel characteristics occurring during transmission (Decision directed mode). The back propagation learning algorithm can be applied directly to the multi-layer perceptron with decision feedback signal [62] as shown in equations (4.10) and (4.11).

5.3 Eliminating Intersymbol Interference: Decision Feedback Signal

The output, $v_j^{(1)}$, of the j^{th} neuron in layer one can be expressed in terms of $\{ g_p \}$, the feedback tap weights $\{ w_{pj}^{b(1)} \}$ and the transmitted signal $\{ u_n \}$ $(u_n \in (1,-1))$ as shown in Figure (5.3). To simplify the mathematics, we assume $d = 0$ in Figure (5.3). Note that $\{ g_p \}$ is the convolution of the channel impulse response $\{ h_p \}$ and the weights $\{ w_{ij}^{(1)} \}$. Thus

$$v_j^{(1)} = f_j\Big(\sum_p u_{n-p}\, g_p + \sum_{p=1}^{l} w_{pj}^{b(1)}\, \tilde{u}_{n-p} + \eta_n + I_j\Big) \tag{5.11}$$

where η_n is zero-mean, Gaussian noise.

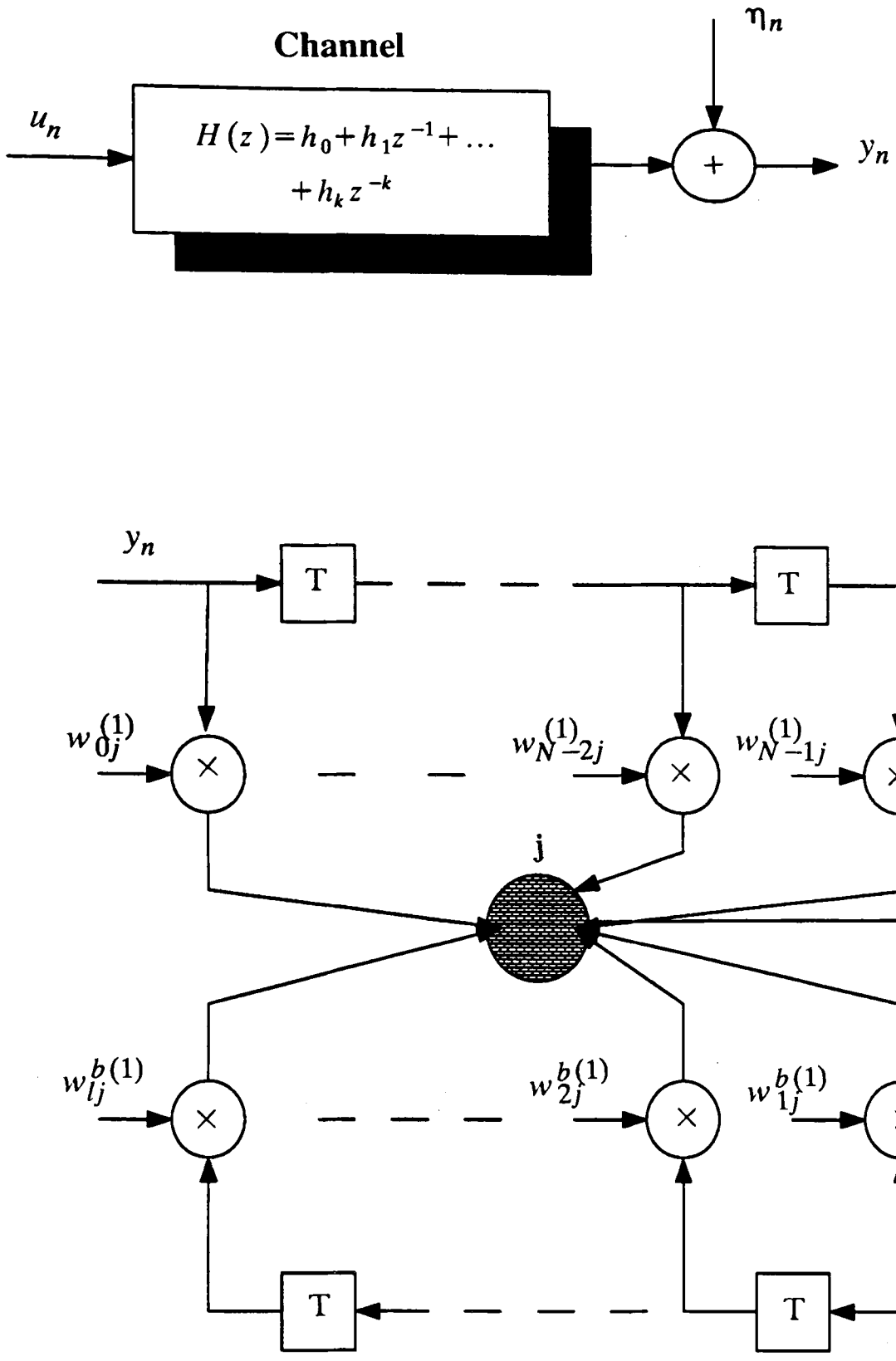


Figure (5.3) $j-th$ neuron with feedback signals in first layer

The above equation can be written as

$$v_j^{(1)} = f_j(u_n\ g_0 + \sum_{p=1}^{l}(g_p + w_{pj}^{b(1)})\tilde{u}_{n-p} + \sum_{p=1}^{l}(u_{n-p} - \tilde{u}_{n-p})\ g_p + \sum_{\substack{p<0 \\ p>l}} u_{n-p}\, g_p + \eta_n + I_j) \tag{5.12}$$

If we select

$$g_p = -w_{pj}^{b(1)} \qquad p = 1,2,.....l \tag{5.13}$$

and the probability of error is very small, we may assume that the last l symbols have been received correctly i.e,

$$\tilde{u}_{n-p} = u_{n-p} \qquad p = 1,2,.....l \tag{5.14}$$

Then equation (5.14), can been simplified as

$$v_j^{(1)} = f_j(u_n\ g_0 + \sum_{\substack{p<0 \\ p>l}} u_{n-p}\ g_p + \eta_n + I_j) \tag{5.15}$$

All ISI from past symbols ($1 \le p \le l$) is eliminated without altering the useful signal term $u_n g_0$ or enchancing the noise component η_n . The $\sum_{\substack{p<0 \\ p>l}} u_{n-p}\, g_p$ residual ISI term will be reduced in the feedforward equalizer. If an incorrect decision is made by the detector i.e. $\tilde{u}_{n-p} = -u_{n-p}$, the decision errors tend to propagate because they result in residual intersymbol interference and a reduced margin against noise at future decisions.

5.4 Perceptron-based DFE Performance and comparison with LMS DFE

The channel model used in the performance evaluation is given in z-transform notation by

$$H(z) = 0.3482 + 0.8704\ z^{-1} + 0.3482\ z^{-2} \tag{5.16}$$

The digital message applied to the channel was in random bipolar form {-1,1}. The channel output is corrupted by zero mean white Gaussian noise. For mathematical convenience, we normalize the received signal power to unity. Then the received signal to noise ratio (SNR) is simply the reciprocal of the noise variance at the input of the equalizer.

The performance was determined by taking an average of 800 individual runs. Each run had a different random sequence, the final bit error rate after convergence was measured and the starting weights were random. The sigmoid function in equation (3.3) was chosen for performance simulation.

For simplicity the short hand notation { (N,l)DFE with (N_1,N_2,N_3)MLP } (MLP DFE) will be used to indicate that the number of received signal samples is N, the number of decision feedback samples is l, the number of neurons in hidden layer 1 (H_1) is N_1, the number of neurons in hidden layer 2 (H_2) is N_2, and the number of neurons in the output layer is N_3, for a three layer perceptron based decision feedback equalizer.

A LMS decision feedback equalizer (LMS DFE) is the same structure as shown in Figure (2.4), and the short hand notation is { (N,l) LMS DFE structure }, where N, and l are both defined as above.

5.4.1 Convergence Characteristics

Figure (5.4) illustrates MSE (mean square error) convergence of the MLP DFE, { (4,1)DFE with (9,3,1)MLP structure }, with learning gain (η) 0.07 and the LMS DFE with learning gain (μ) 0.035. For illustrative purposes the MSE convergence of equalizers for no feedback signal are shown. The MLP DFE requires at least 1000

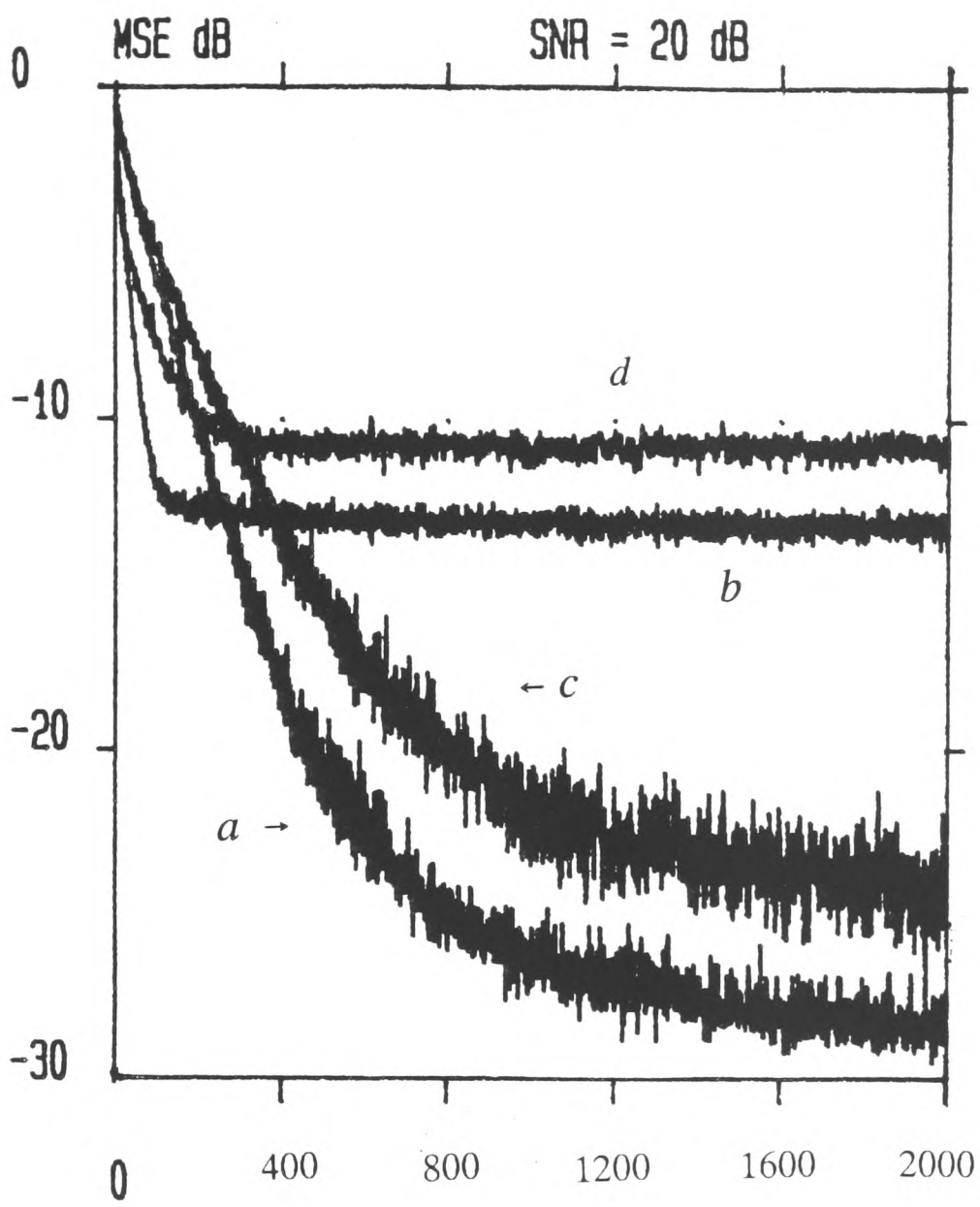


Figure (5.4) Simulation results showing relative convergence rate performance for SNR = 20 dB

(a) {(4,1)DFE with (9,3,1)MLP structure}: $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$

(b) {(4,1)DFE LMS structure}: $\mu = 0.035$

(c) { (9,3,1)MLP structure with $N = 5$ }(without feedback signal) : $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$

(d) { LMS equalizer structure with $N = 5$ }(without feedback signal) : $\mu = 0.035$

iterations to converge while the LMS DFE converges in about 120 iterations. The results also show that the steady-state value of averaged square error produced by the MLP DFE converge to a value (<-25 dB) which is lower than the additive noise (-20 dB). This is due to the nonlinear nature of equalizer transfer function. The LMS DFE gives a steady value of averaged square error at about -14.0 dB which is above the noise floor using the the same number of input samples. The result also indicates that both types of the decision feedback equalizer yield a significant improvement in convergence time and averaged square error relative to the equalizers without feedback signal having the same number of input samples.

5.4.2 Bit error rate performance - Decision directed mode

DFE performance can be obtained by means of a Monte Carlo simulation. Figure (5.5) illustrates the bit error rate performance in the stationary channel case for the MLP DFE performance using either correct or detected symbols in the feedback section with $\eta = 0.1$. For illustrative purposes the performance of the perceptron based equalizer (without feedback) with $\eta = 0.1$ is shown.

It may be observed from Figure (5.5) that the MLP DFE attains about 4 dB improvement at BER = 10^{-4} relative to the MLP equalizer having the same number of input samples. The performance loss due to incorrect decisions being fed back is 1.3 dB, approximately, for the channel response under consideration.

Figure (5.6) illustrates the performance of the LMS DFE with $\mu = 0.035$. The results show the LMS DFE attains 4 dB improvement at BER = 10^{-4} relative to the LMS equalizer. The performance loss due to incorrect decisions being fed back is about 2.0 dB. From the data in Figures (5.5) and (5.6) can be seen that the perfor-

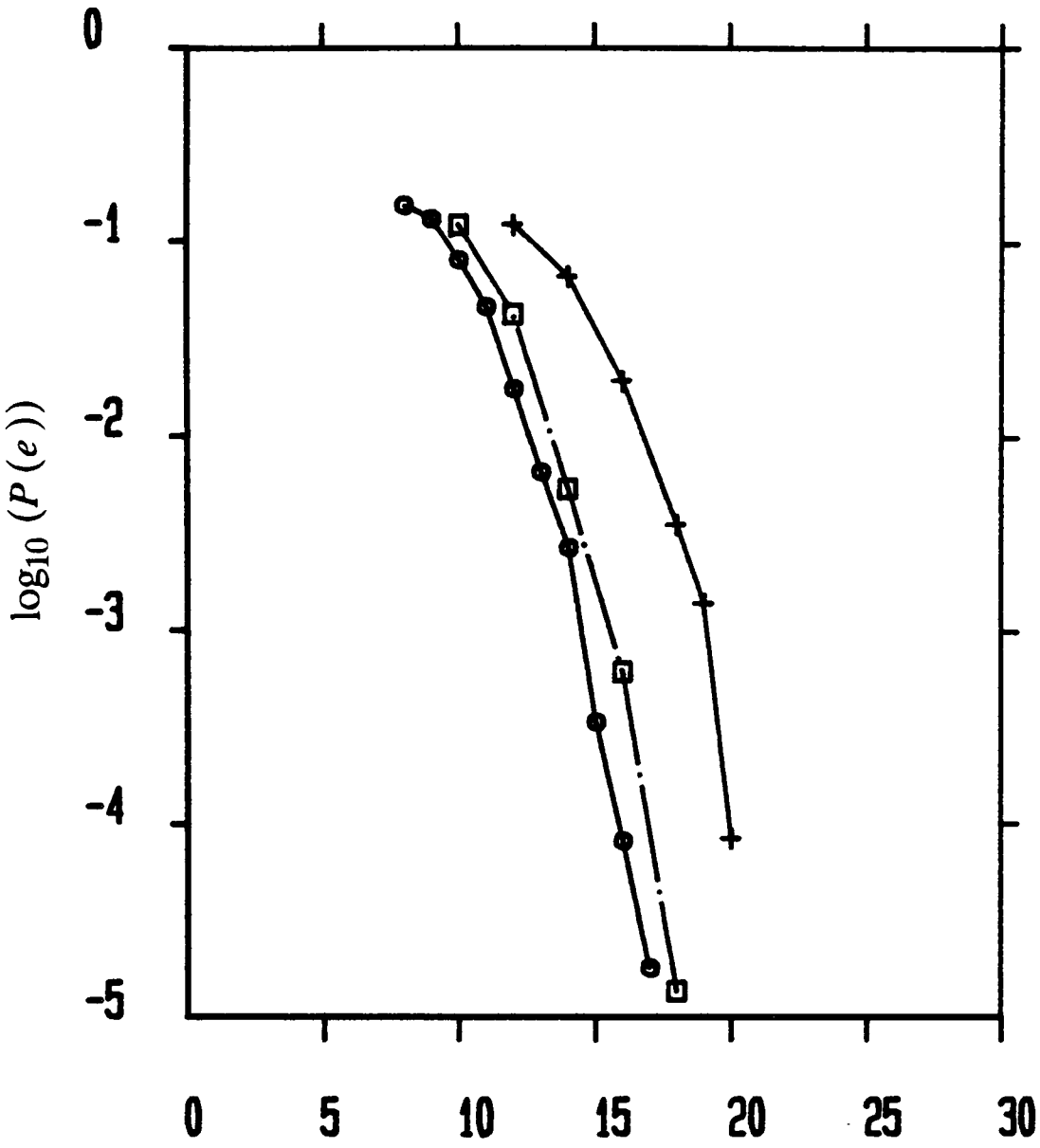


Figure (5.5) Bit error rate performance of {(4,1)DFE with (9,3,1)MLP structure } with and without error propagation, η = 0.1, α = 0.3, β = 0.05.

o : Correct bit fed back,

□ : Detected bit fed back.

+ : { (9,3,1)MLP structure with $N = 5$ }(without feedback signal).

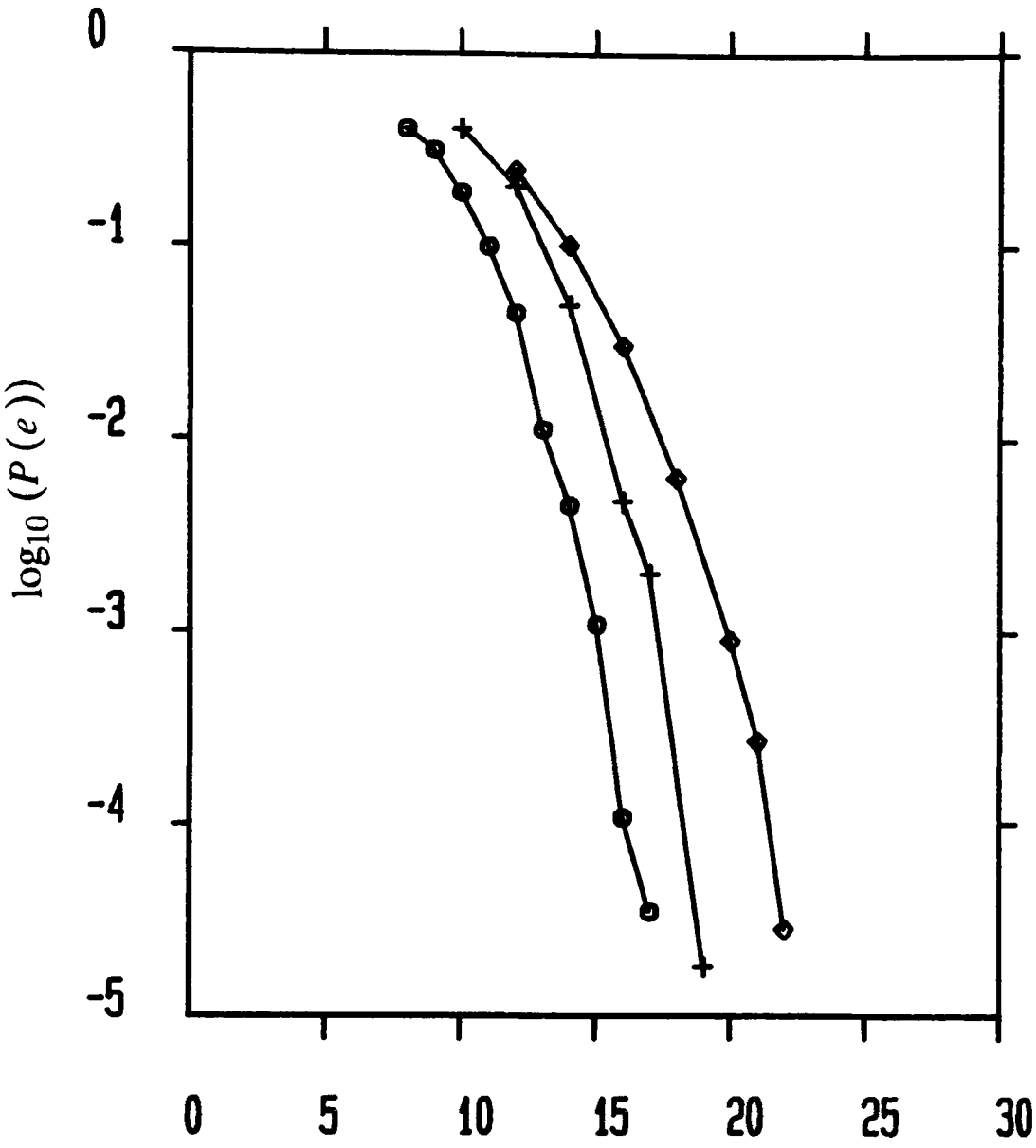


Figure (5.6) Bit error rate performance of {(4,1)DFE LMS structure} with and without error propagation, $\mu = 0.035$.

o : Correct bit fed back,

+ : Detected bit fed back.

◇ : { LMS equalizer structure with $N = 5$ }(without feedback signal): $\mu = 0.035$.

mance degradation due to decision errors for the perceptron-based DFE is less than for the LMS DFE, especially under high noise conditions.

Figure (5.7) illustrates performance for the MLP DFE with η = 0.07, and 0.1 compared with the LMS DFE with μ = 0.035, and 0.05 having the same number of input samples. In the simulation, all the symbols fed back are detected symbols. The MLP DFE structure has superior performance in comparison with the LMS DFE structure , when the level of additive noise is high, but the improvement reduces as the signal to noise ratio improves. This latter fact should not surprise us since, if the additive noise level is very low, the MLP DFE will receive very few samples of signal which are close to the optimal decision boundary, rendering it incapable of having the optimal decision boundary as it does in the high noise situation.

5.5 Discussion

The conventional structure of the DFE consists of a feedforward equalizer and a feedback filter, where the feedforward equalizer is linear. The linearity of the feedforward equalizer limits the performance of the DFE.

This chapter has introduced a new approach for the DFE using multi- layer perceptron structures. The back propagation learning algorithm is applied directly to the multi-layer perceptrons. From comparison of simulation results it can be seen that the multi-layer perceptron-based DFE provides better BER performance, especially in poor signal to noise ratio conditions, also that BER performance degrades less due to decision errors and is also less sensitive to gain variation.

The bit error rate performance of multi-layer perceptron-based DFE is near that of LMS DFE at " high " SNR, while the learning gain (μ) in LMS algorithm is very

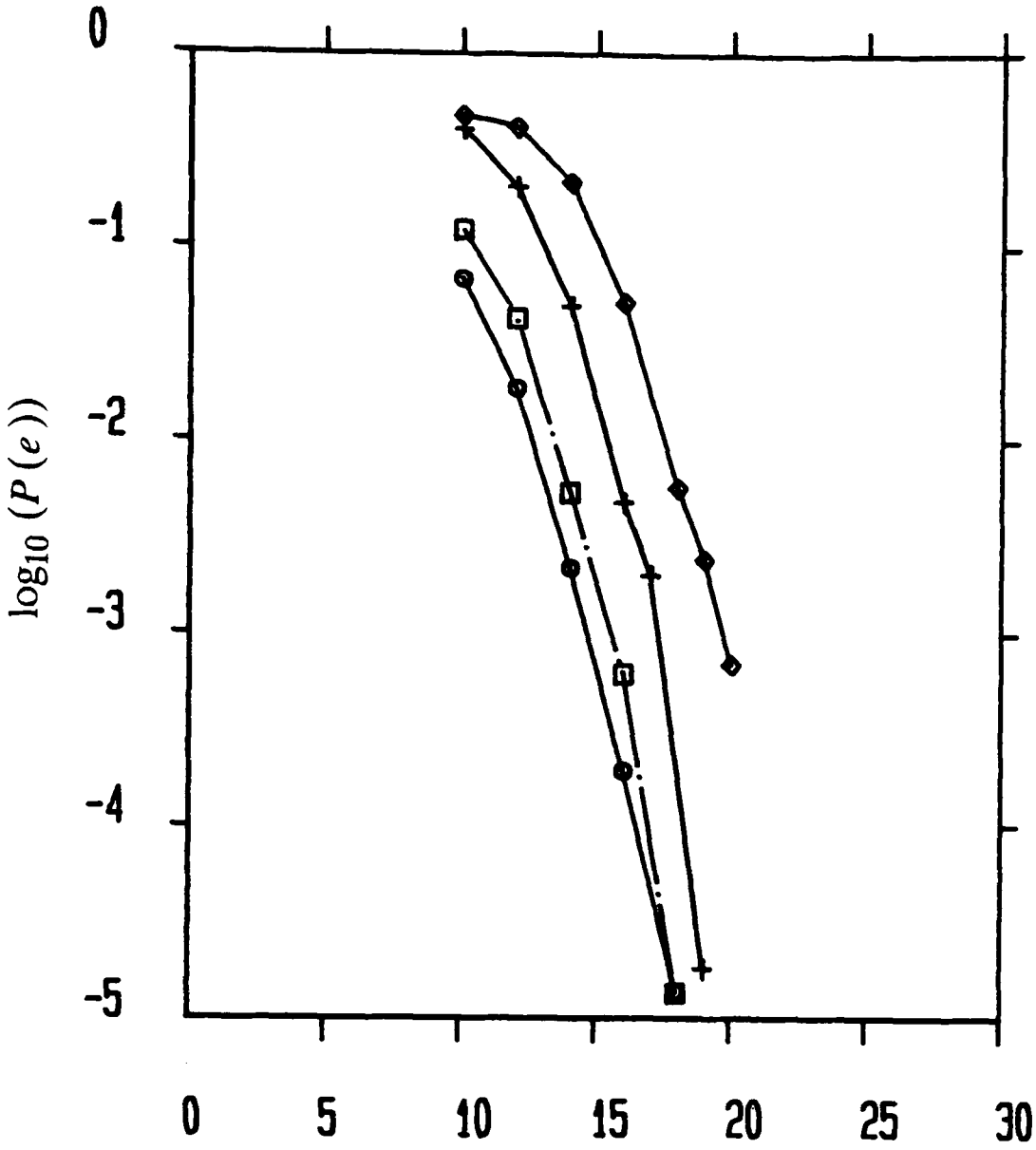


Figure (5.7) Simulation results showing relative bit error rate performance for {(4,1)DFE with (9,3,1)MLP structure}:

□ : $\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$,

o : $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$.

{(4,1)DFE LMS structure}:

◇ : $\mu = 0.05$,

+ : $\mu = 0.035$.

small (see sections (5.4.2) and (4.3.4))

It may be concluded that the multi-layer perceptron-based DFE offers a superior performance (higher resolution) as a channel equalizer to that of the conventional DFE , due to its ability to form complex decision regions with nonlinear boundaries.

CHAPTER 6

MINKOWSKI- ζ BACK PROPAGATION

AND

LEARNING PARAMETER CONSIDERATIONS

6.1 Introduction

A typical estimation problem is illustrated in Figure (6.1), where $h(\cdot)$ is some specified (linear or nonlinear) function operating on an observed random sequence y_n, to yield an estimate, $\hat{u}_n$ of u_n (desired signal). The error in the estimate is :

$$e_n = u_n - \hat{u}_n \tag{6.1}$$

The purpose of the estimation problem is to try to choose $h(\cdot)$ so as to minimize the expected (average) value of some cost function. Some typical cost functions [20],[63],[64] are shown in Figure (6.2). With the squared-error function, $|e|^2$, small errors have less emphasis than larger errors due to the nonlinear square function; the absolute error function, $|e|$, gives equal weight to all errors; and in the uniform cost function, $f(e)$, errors less than κ in magnitude incur no penalty. The mean squared function is usually chosen due to its continuously differentiable nature. In this chapter a new model of back propagation is discussed using the cost function, $|e|^{\zeta}$,

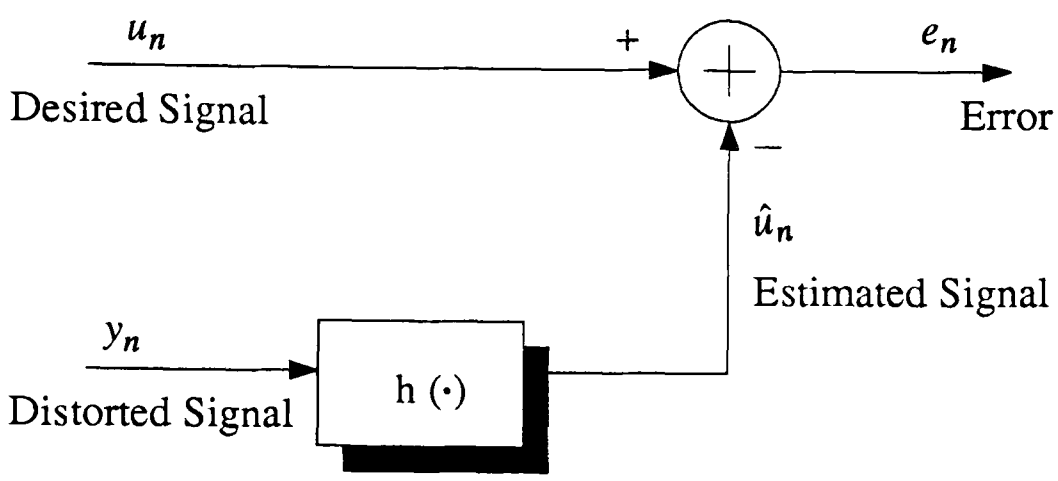


Figure (6.1) Optimal Estimation

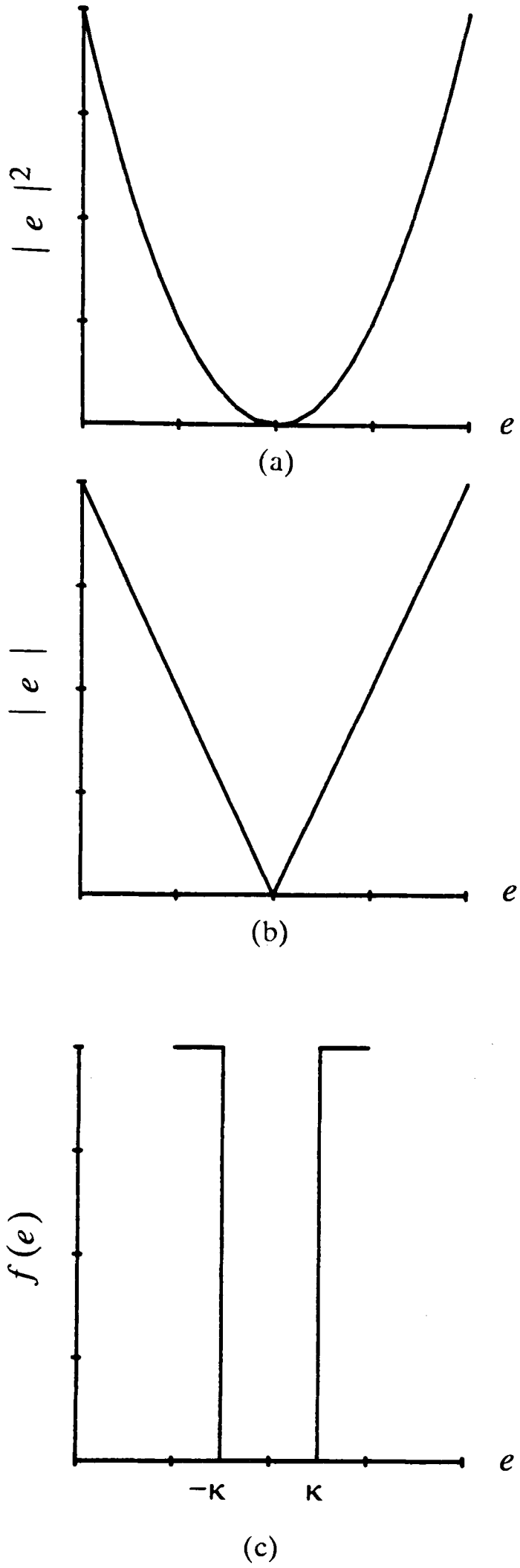


Figure (6.2) Cost function (a) Square error, (b) Absolute error, and (c) Uniform cost function.

where $1 \leq \zeta \leq 2$ (Minkowski-ζ power metrics) [65] and some of its properties. In particular, the effect of ζ on the performance (e.g., convergence time, bit error rate, and mean square error) is explored together with robustness.

The stochastic gradient algorithm or LMS [66] has only a single learning parameter for controlling the convergence rate, namely, the learning gain μ . The back propagation algorithm (equations (4.10) and (4.11)) has three learning parameters, namely, the learning gain η, the momentum parameter α, and the threshold level gain β which make the choice of parameters much more complicated. Furthermore the performance (convergence rate and bit error rate) is determined particularly by the η and α parameters. The effect of η and α parameters on the performance is analysed in this chapter.

6.2 **Minkowski-ζ** Back Propagation

The standard back propagation learning algorithm is based on the squared error cost function. The present model is a variation on the model of back propagation using Minkowski-ζ power error metrics [65],[67]. The error cost function for Minkowski-ζ power metrics is given by

$$E = \frac{1}{\zeta} \sum_j (|t_j - v_j^{(M)}|)^\zeta \tag{6.2}$$

where ζ($\zeta > 0$), while for $\zeta=2$ the standard back propagation model results. The Minkowski-ζ family E (error) versus the factor ($|t_j - v_j^{(M)}|$) for different values of ζ is shown in Figure (6.3).

Using the chain rule, the error signal in the output layer in equation (3.18) can be rewritten as

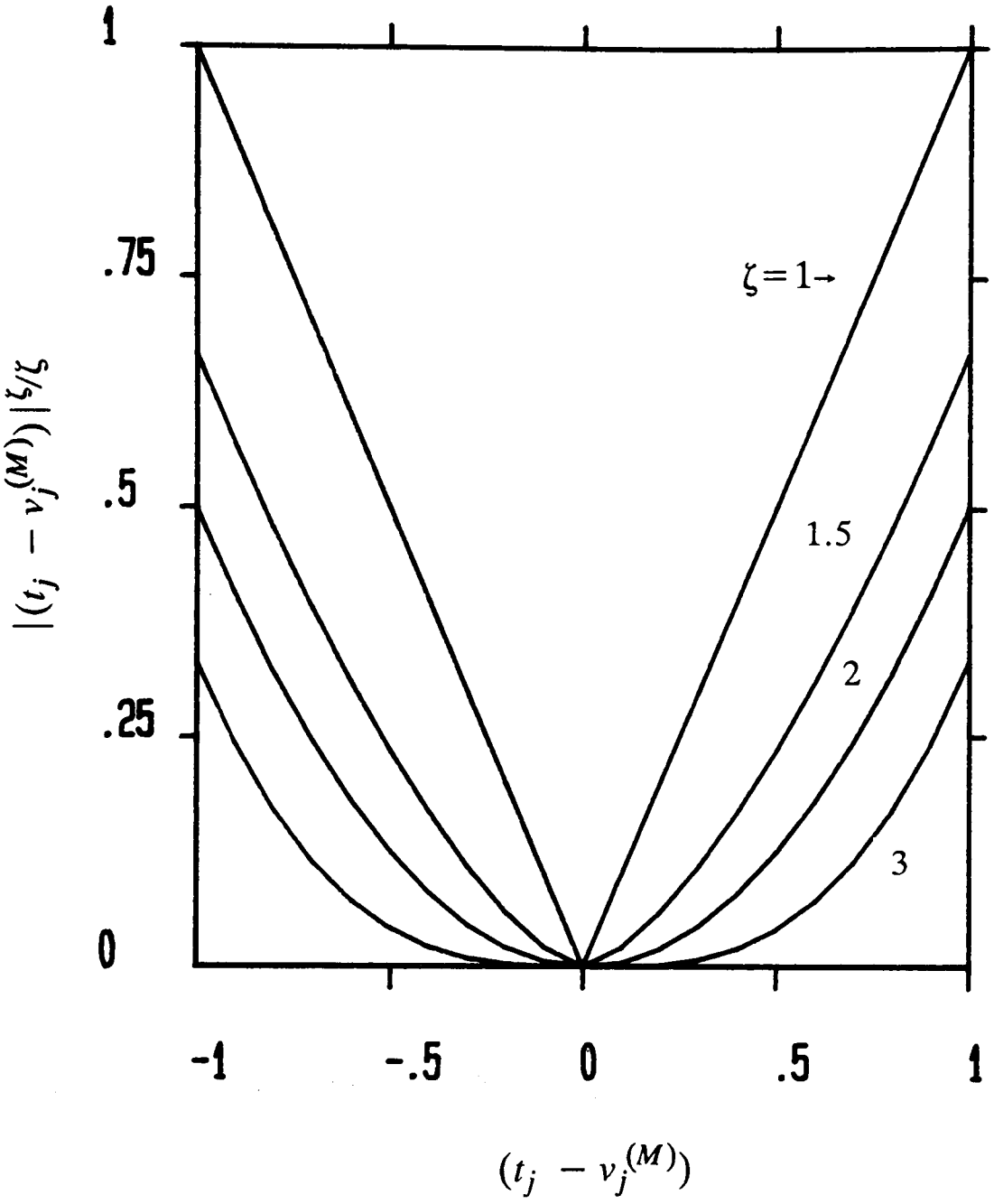


Figure (6.3) Minkowski- ζ family

$$\delta_j^{(M)} = -\frac{\partial E}{\partial v_j^{(M)}} \frac{\partial v_j^{(M)}}{\partial x_j^{(M)}} \tag{6.3}$$

The sigmoid function is given by

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{6.4}$$

Since $v_j^{(M)} = f_j(x_j^{(M)})$ (see equation(3.3)) , the derivative of $v_j^{(M)}$ is then

$$\frac{\partial v_j^{(M)}}{\partial x_j^{(M)}} = (1 - v_j^{2\,(M)})/2 \tag{6.5}$$

Substituting equations (6.2), and (6.5) into equation (6.3), the error signal in the output layer is then

$$\delta_j^{(M)} = (|t_j - v_j^{(M)}|)^{\zeta-1} (1-v_j^{2\,(M)})/2 \; sign(t_j - v_j^{(M)}) \tag{6.6}$$

where

$$sign(\psi) = \begin{cases} 1 & \text{if } \psi > 0 \\ -1 & \text{if } \psi < 0 \end{cases} \tag{6.7}$$

$$(|t_j - v_j^{(M)}|)^{\zeta-1} \geq (|t_j - v_j^{(M)}|) \qquad \zeta < 2 \tag{6.8}$$

$$(|t_j - v_j^{(M)}|)^{\zeta-1} \leq (|t_j - v_j^{(M)}|) \qquad \zeta > 2 \tag{6.9}$$

The following illustrates (equation (6.6)) for $\zeta = 1,\ 1.2,\ 2$.

$$\delta_j^{(M)} = \begin{cases} (1-v_j^{2\,(M)})/2 \; sign(t_j - v_j^{(M)}) & \zeta = 1 \\ (|t_j - v_j^{2\,(M)}|)^{0.2} (1-v_j^{2\,(M)})/2 \; sign(t_j - v_j^{(M)}) & \zeta = 1.2 \\ (t_j - v_j^{(M)})(1-v_j^{2\,(M)})/2 & \zeta = 2 \end{cases} \tag{6.10}$$

The rule for weight updating proceeds in the same way as in the standard back propagation (equation(3.22)) except there is no momentum term, where

$$\Delta w_{ij}^{(m)}(n+1) = \eta \delta_j^{(m)}(n)\; v_j^{(m-1)}(n) \tag{6.11}$$

The error signal in the hidden layer is a function of the error signal in the output, $\delta_j^{(M)}$ and is the same as equation (3.19), where

$$\delta_j^{(m)} = (1 - v_j^{2\,(m)}) \sum_l \delta_l^{(m+1)} w_{lj}^{(m+1)} /2 \tag{6.12}$$

where $m \in \{1,2,\ldots, M-1\}$ and l is over all neurons in the layer above neuron j.

The Minkowski-ζ Back propagation using another type of sigmoid function, $f(x)=(1+e^{-x})^{-1}$ is also shown in Table 6.1.

6.3 Properties of Minkowski- ζ Back Propagation

Equation (6.6) indicates that changing ζ basically results in a reweighting of errors $(t_j - v_j^{(M)})$ from the output in $\delta_j^{(M)}$. The factor $(|t_j - v_j^{(M)}|)^{\zeta-1}$ is nonlinear when $\zeta \neq 2$ as this introduces the nonlinearity in the output error signal, $\delta_j^{(M)}$ for the Minkowski-ζ back propagation. Small ζ's give less weight for large deviations of $(t_j - v_j^{(M)})$ in $\delta_j^{(M)}$ and tend to reduce the influence of the aberrant noise. In contrast to power value (ζ) equal to 2 (standard back propagation), which gives equal weight to all errors $(t_j - v_j^{(M)})$ in $\delta_j^{(M)}$. Further, decreasing ζ can significantly improve convergence because the error signal $\delta_j^{(M)}$ is enhanced by the factor $(|t_j - v_j^{(M)}|)^{\zeta-1}$ and vice versa, as shown in equations (6.6). Figure (6.4) depicts the factor $(|t_j - v_j^{(M)}|)^{\zeta-1}$ versus $(|t_j - v_j^{(M)}|)$ for different values of ζ.

In summary, values of ζ close to 1 the error signal $\delta_j^{(M)}$ is less sensitivity to large deviation of $(t_j - v_j^{(M)})$ than when ζ is closer to 2. If ζ is allowed to increase beyond 2, $\delta_j^{(M)}$ is more sensitive to the large deviation of $(t_j - v_j^{(M)})$. When $\zeta < 2$, the tracking capability of the system will be enhanced more significantly for small deviations of $(|t_j - v_j^{(M)}|)$. As a result a lower value of steady-state mean square error will be

Table (6.1) Error Feedback Signal

Sigmoid type	$f(x)=\frac{1}{1+e^{-x}}$
Output layer	$\delta_j^{(M)}=(\lvert t_j-v_j^{(M)}\rvert)^{\zeta-1}v_j^{(M)}(1-v_j^{(M)})sign(t_j-v_j^{(M)})$ $1\leq \zeta \leq 2$
Hidden layer	$\delta_j^{(m)}=v_j^{(m)}(1-v_j^{(m)})\sum_l \delta_l^{(m+1)}w_{lj}^{(m+1)}$ $m \in [1,2,\ldots.M-1]$

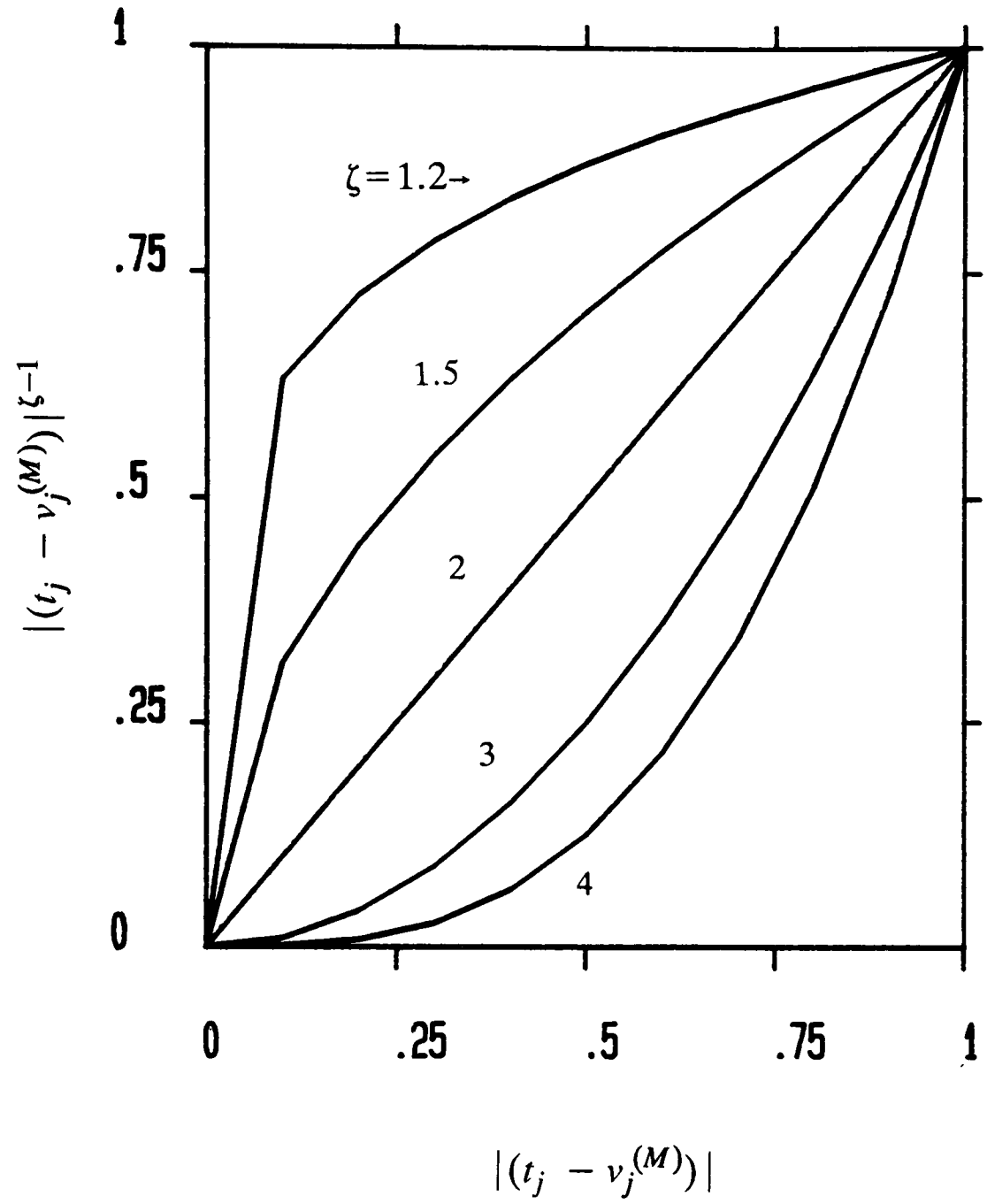


Figure (6.4) $|(t_j - v_j^{(M)})|^{\zeta-1}$ vs. $|(t_j - v_j)|$ for different values of ζ.

achieved in comparison with $\zeta = 2$.

6.4 Effects of Minkowski-ζ power metrics on the performance

The performance of multi-layer perceptrons trained with back propagation using Minkowski-ζ power metrics, where $1<\zeta\leq 2$, was examined using the perceptron based equalizer scenario { (9,3,1)MLP structure with $N=5$ } with different values of learning gain (η) and no momentum term ($\alpha=0$). Throughout, the nonminimum phase channel in equation (4.14), and the sigmoid function in equation (3.3) were chosen for performance simulation. The simulations provided in this section confirm the results described in section (6.3).

6.4.1 Convergence Properties

In general, the mean squared error (MSE) is one of the most useful measures equalizer performance. This technique will be used in the section. Figure (6.5) shows the convergence time to the noise floor as a function of the Minkowski-ζ power metrics for 800 different initial conditions (random starting weights). The result indicates that as ζ decreases, convergence time tends to improve roughly linearly. The Minkowski-ζ back propagation with $\zeta<2$ was able to reduce the steady-state of the mean square error more than the standard back propagation with $\zeta=2$, as shown in Figure (6.6). The mean square error after convergence shows approximately 10 dB improvement as ζ decreases from 2 to 1.2 for $\eta = 0.2$. This is because the error $(t_j - v_j^{(M)})$ in the output $\delta_j^{(M)}$ is enchanced by the term $(|t_j - v_j^{(M)}|)^{\zeta-1}$ which improves the tracking capability of the network.

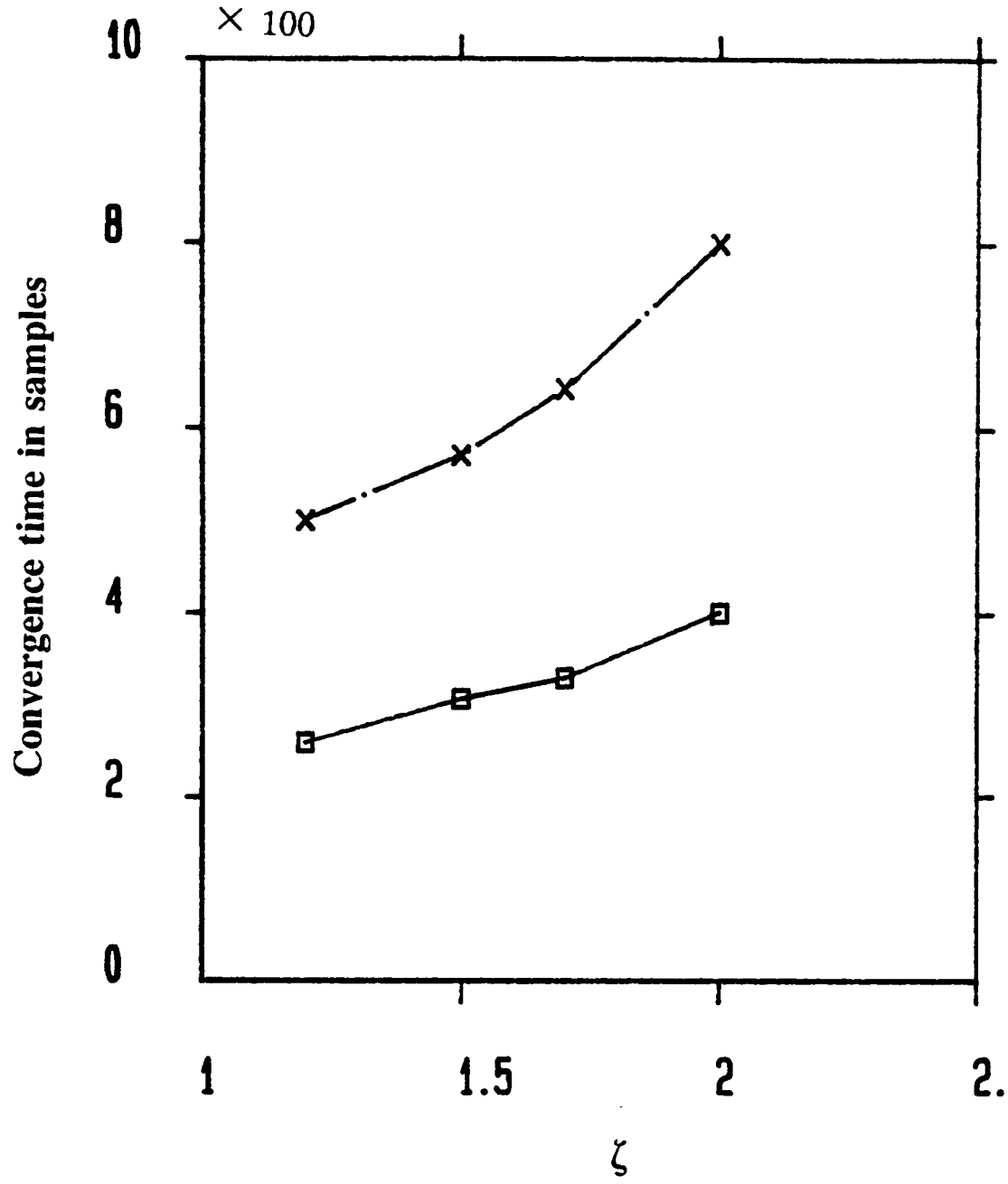


Figure (6.5) The convergence time to the noise floor (-20 dB) for a perceptron-based equalizer, { (9,3,1)MLP structure with N=5 } as a function of ζ at SNR = 20 dB.

× : η = 0.1, α = 0.0, and β = 0.05.

□ : η = 0.2, α = 0.0, and β = 0.05.

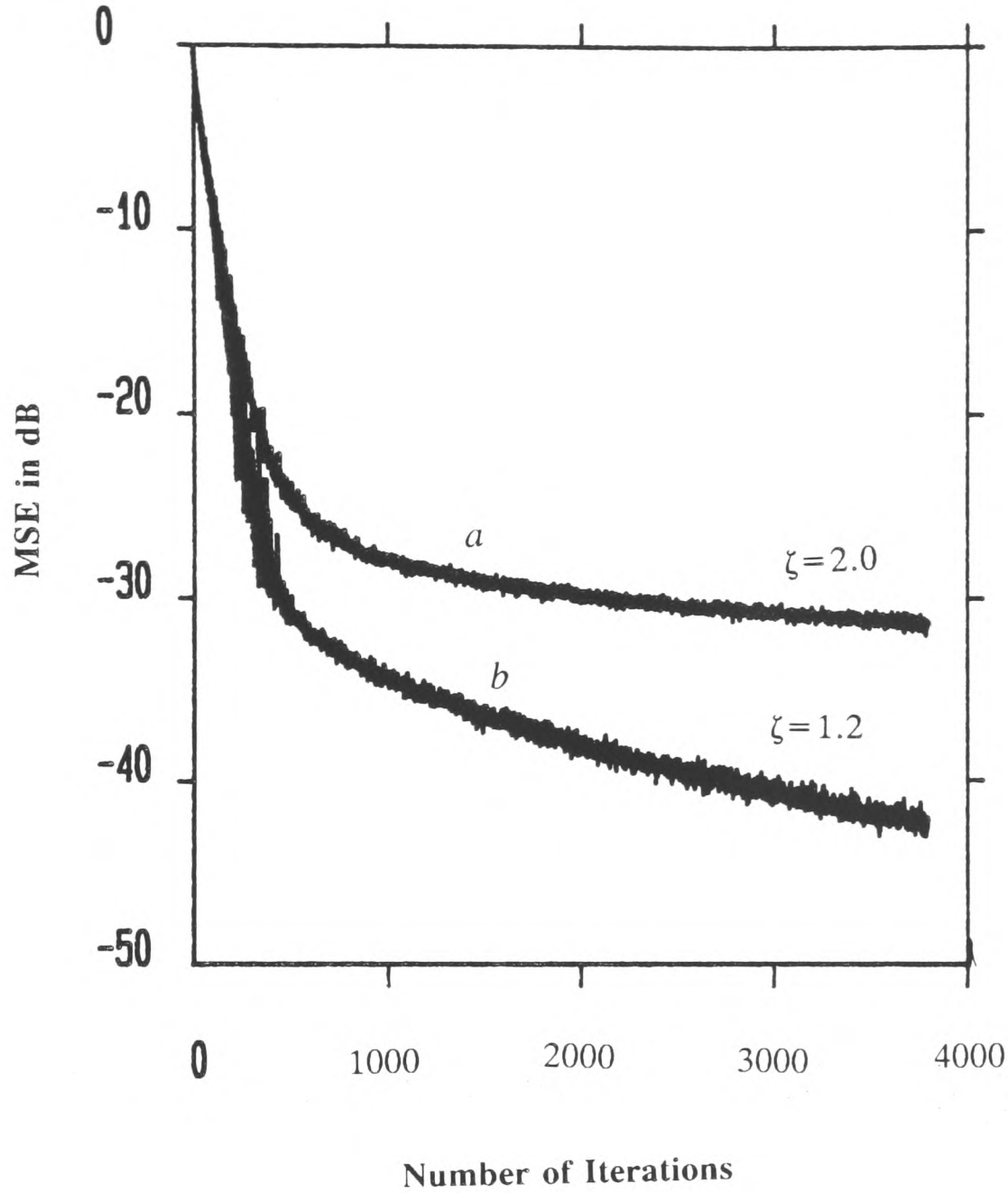


Figure (6.6) Simulation results showing relative convergence rate performance for a perceptron-based equalizer, { (9,3,1)MLP structure with N=5 } at SNR = 40 dB.

(a) $\eta = 0.2$, $\alpha = 0.0$, $\beta = 0.05$, and $\zeta = 2.0$.

(b) $\eta = 0.2$, $\alpha = 0.0$, $\beta = 0.05$, and $\zeta = 1.2$.

6.4.2 Bit Error Rate Performance-Decision-Directed Mode

Figures (6.7), and (6.8) illustrate the bit error rate performance versus Minkowski-ζ power metrics with SNR = 18 and 20 dB for η = 0.1 and 0.2, respectively. The result indicates that as ζ decreases, a lower bit error rate can be achieved. The improvement is more significant using a 20 dB SNR and η = 0.2, and the bit error rate is reduced by about 10 times as ζ decreases from 2 to 1.2; when SNR is 18 dB the improvement is only two times. This shpae in the SNR = 20 dB curve in Figure (6.7) shows a peculiar increase in bit error rate at high ζ . this require further investigation.

Figure (6.9) compares the bit error rate performance as a function of signal to noise ratio with ζ= 2, and 1.5. While ζ = 1.5 exhibits better performance, the improvement is more significant in the case of high signal to noise ratio.

Finally, the relative bit error rate performance with ζ = 2.0 and 1.2 as a function of η for SNR = 20 dB was simulated and the results are shown in Figure (6.10). When ζ = 1.2 the results show better performance and the improvement is more significant for larger η.

A more superior bit error rate performance is achieved for $\zeta < 2$ in comparison with ζ = 2. The reason is that small ζ's give less weight for large deviations of $(t_j - v_j^{(M)})$ in $\delta_j^{(M)}$ and tend to reduce the influence of the aberrant noise. In contrast to power value (ζ) equal to 2, which gives equal weight to all errors $(t_j - v_j^{(M)})$ in $\delta_j^{(M)}$. Further the error signal $\delta_j^{(M)}$ is enhanced more significantly for small deviations of $(|t_j - v_j^{(M)}|)$ for $\zeta < 2$. Thus the tracking capability of the system is improved resulting in the steady-state mean square error is to be reduced.

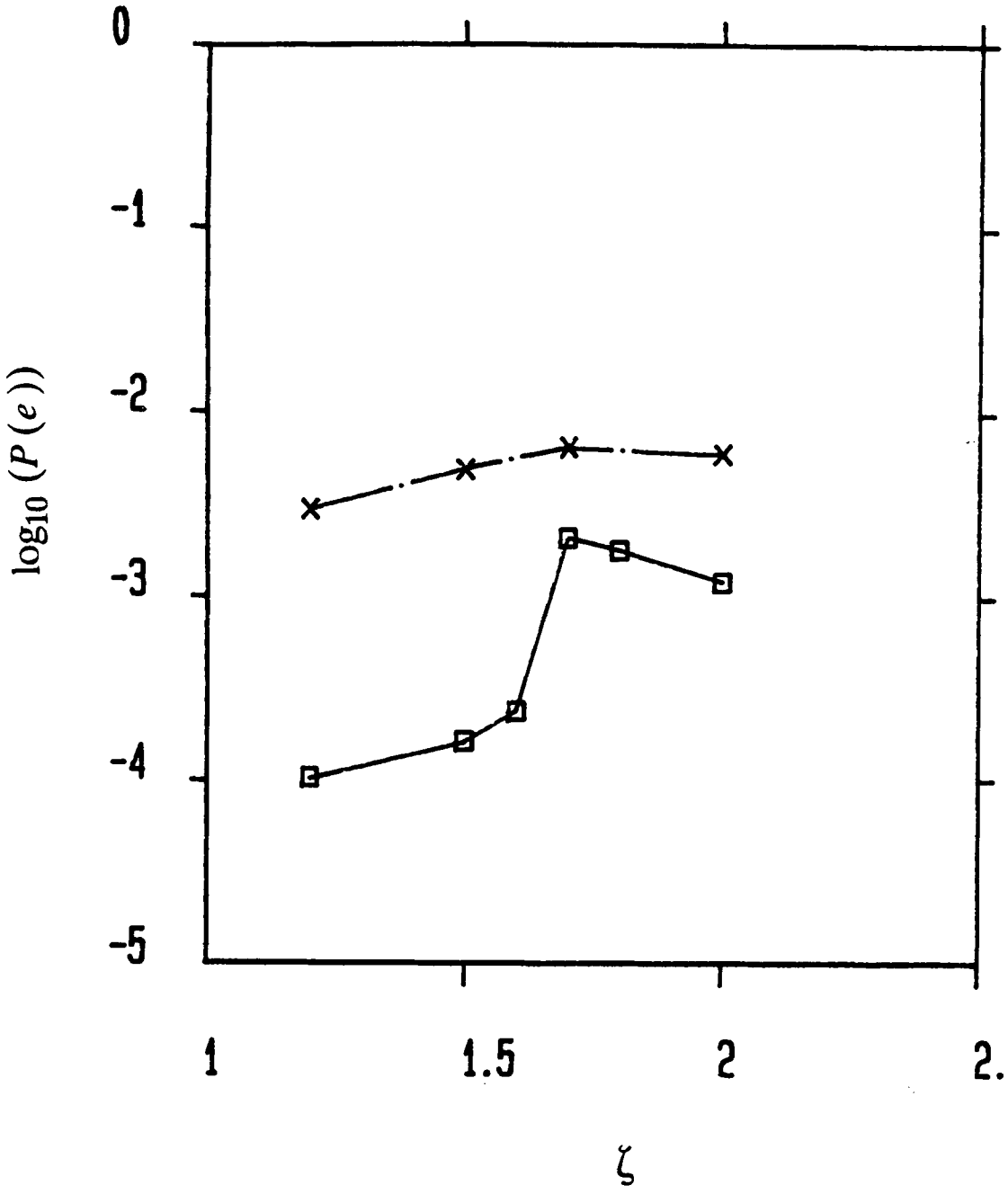


Figure (6.7) The bit error rate performance of the perceptron-based equalizer, { (9,3,1)MLP structure with N=5 } as a function of ζ. Parameters $\eta = 0.2$, $\alpha = 0.0$, and $\beta = 0.05$.

× : SNR = 18 dB,

□ : SNR = 20 dB.

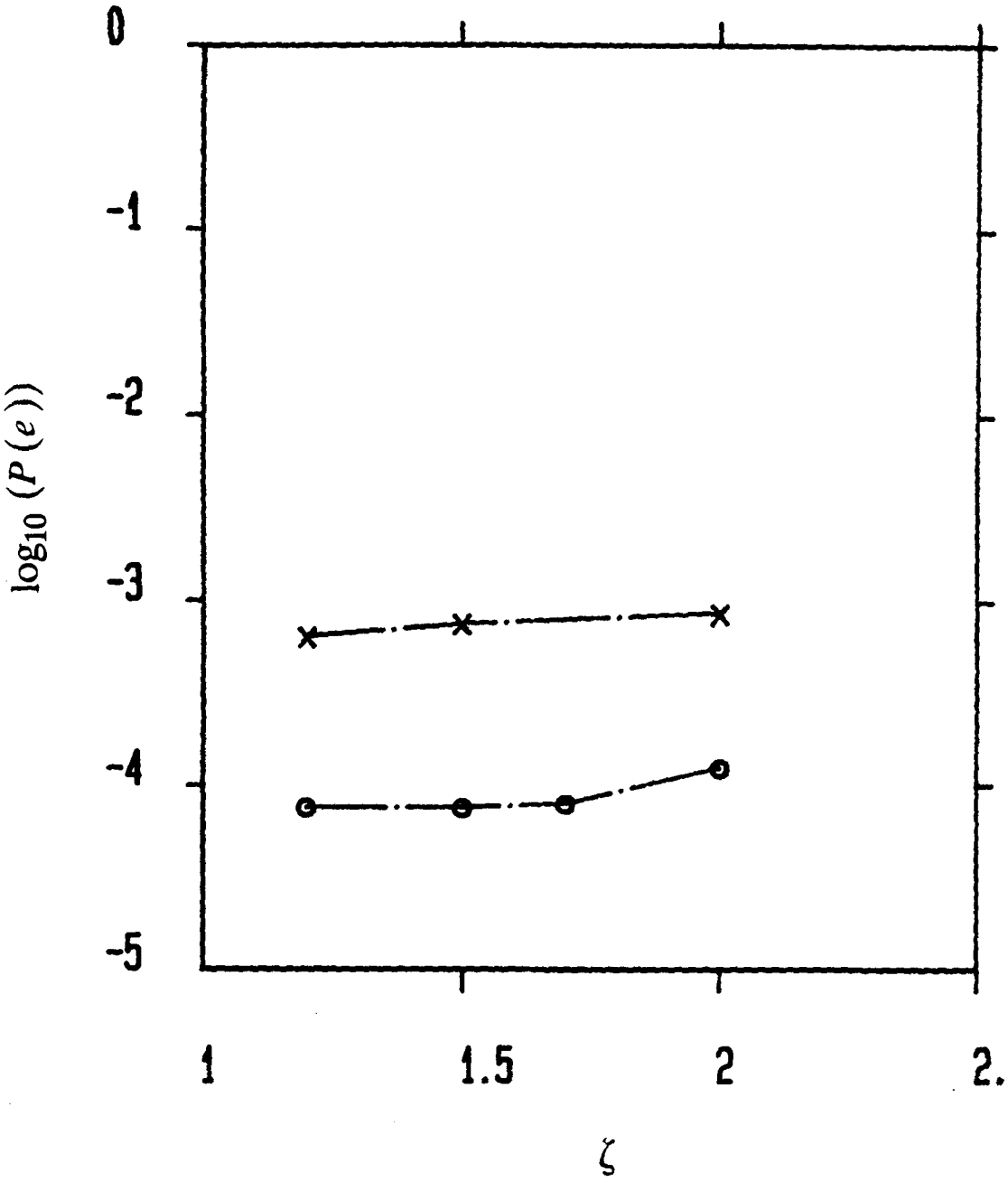


Figure (6.8) The bit error rate performance of the perceptron-based equalizer. { (9,3,1)MLP structure with N=5 } as a function of ζ. Parameters $\eta = 0.1$, $\alpha = 0.0$, and $\beta = 0.05$.

$\times$: SNR = 18 dB,

$\mathfrak{o}$: SNR = 20 dB.

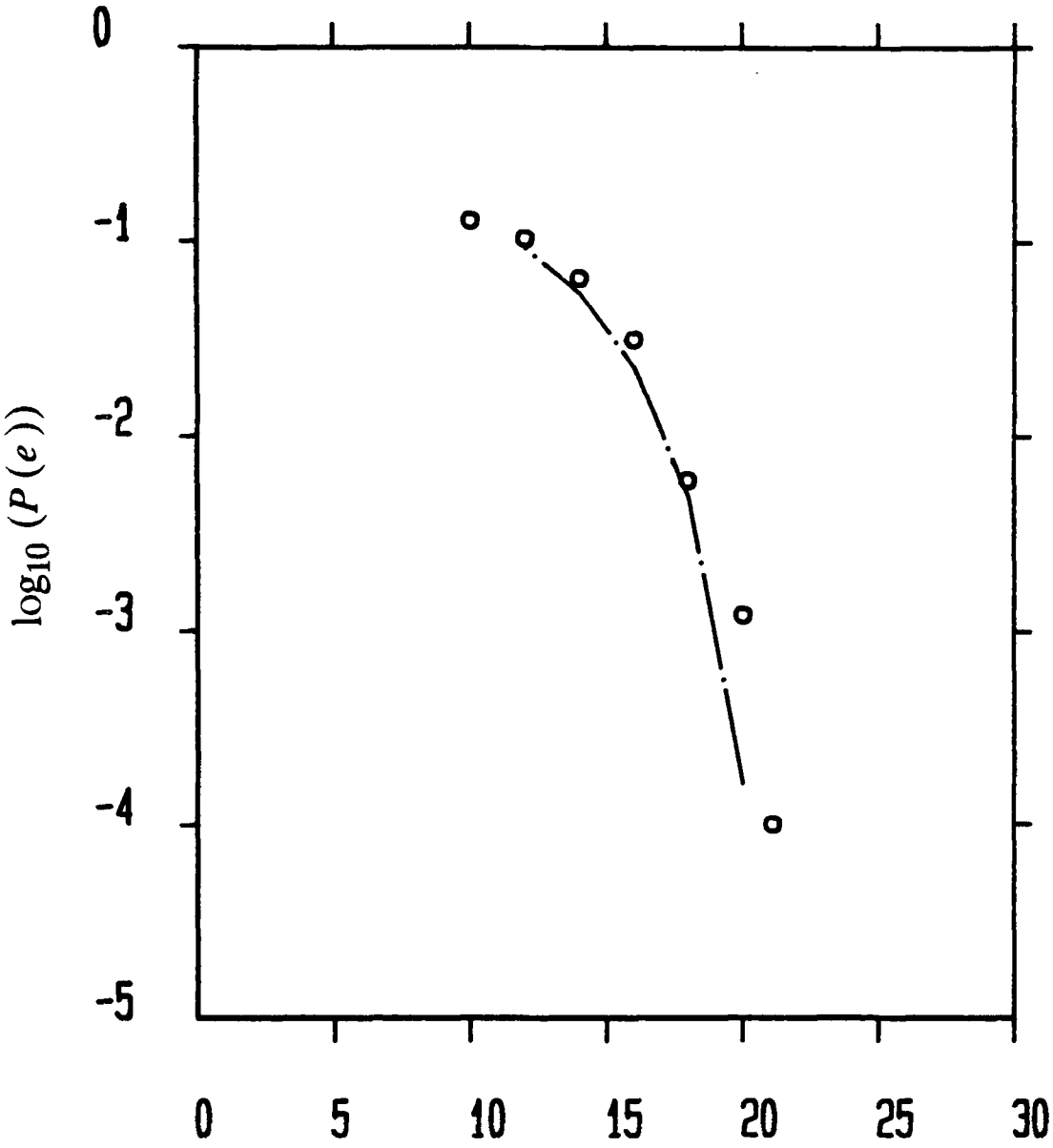


Figure (6.9) The bit error rate performance of the perceptron-based equalizer { (9,3,1)MLP structure with N=5 } as a function of signal-to-noise ratio. Parameters $\eta = 0.2$, $\alpha = 0.0$, and $\beta = 0.05$.

$-\cdot-$: $\zeta = 1.2$,

o : $\zeta = 2$.

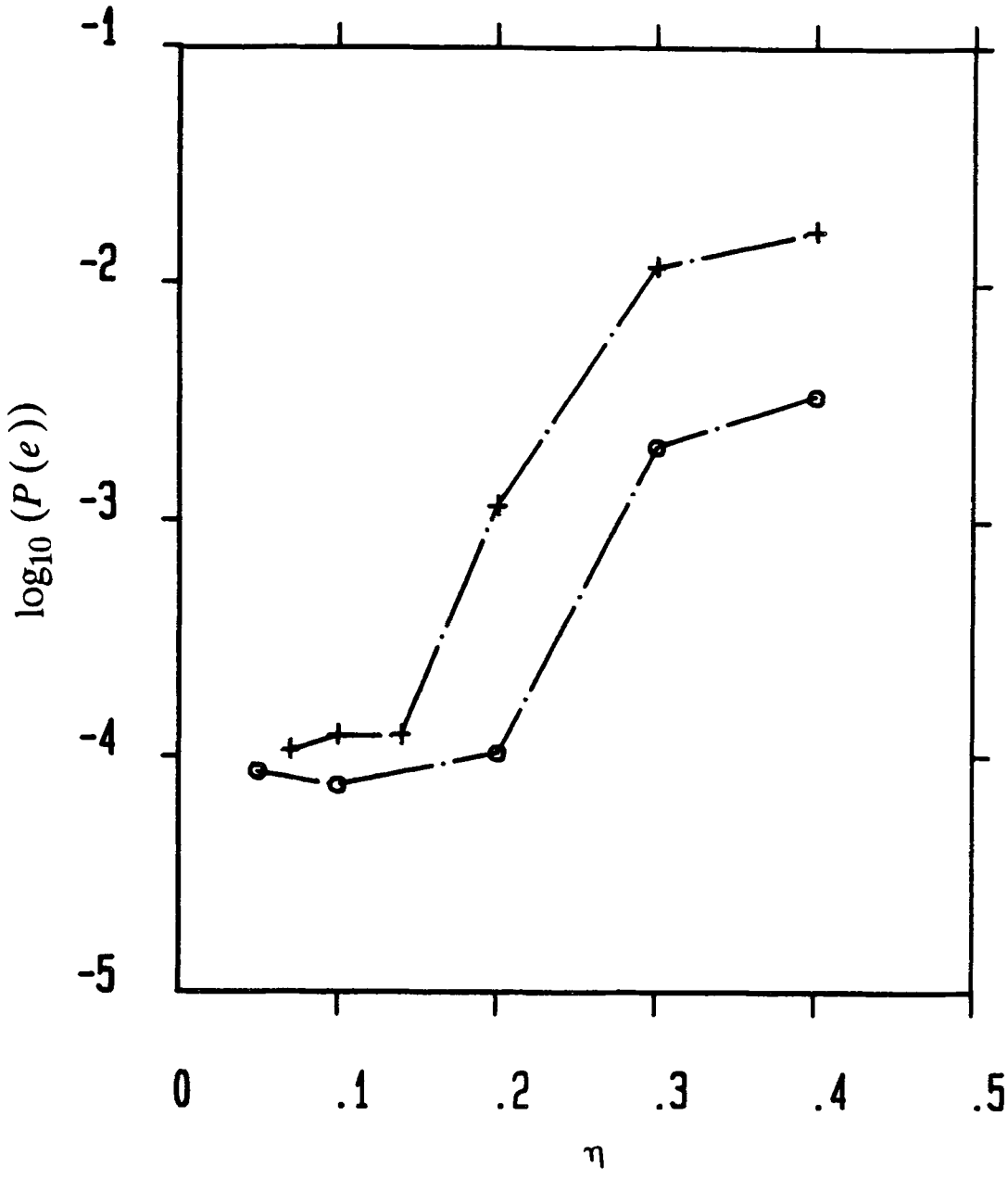


Figure (6.10) Simulation results showing relative bit error rate performance for a perceptron-based equalizer, { (9,3,1)MLP structure with N=5 } with $\zeta = 1.2$, and 2.0 as a function of η for SNR = 20 dB.

+ : $\zeta = 2.0$,

o : $\zeta = 1.2$.

6.5 Learning Parameter Considerations

The performance of the multi-layer perceptron (MLP) trained with the back propagation learning algorithm depends on several factors including network topology, complexity of decision regions, initial weights and choice of learning parameters (learning gains, momentum parameter) in the learning algorithm, and error cost function characteristics.

An effective learning gain that relates the learning gain (η) and the momentum parameter (α) is derived. The analysis result is only approximation, but it is supported by computer simulations. The effect of various learning gains and momentum parameters on the performance (e.g., convergence time, bit error rate) is studied using the equalizer scenario. The result may be useful as guideline for design of multi-layer perceptrons using the back propagation learning algorithm.

6.5.1 Effective Learning Gain

The weight w is updated according to

$$w^{(m)}(n+1) = w^{(m)}(n) + \eta\, \delta^{(m)}(n)\, v^{(m-1)}(n) + \alpha\, (w^{(m)}(n) - w^{(m)}(n-1)) \quad (6.13)$$

where $m \in [1,2,\ldots M]$ and the last term in the right hand side of equation (6.13) is the momentum term which smooths out high frequency variations in the weight during updating. The frequency bandwidth is determined by the momentum parameter " α ".

If the equation (6.13) is iterated from n back to 0, we obtain

$$w^{(m)}(n) = \sum_{i=0}^{n} \alpha^i w^{(m)}(0) + \eta \sum_{i=0}^{n-1} \alpha^i \delta^{(m)}(0)\, v^{(m-1)}(0) + \eta \sum_{i=0}^{n-2} \alpha^i \delta^{(m)}(1)\, v^{(m-1)}(1) + \ldots\ldots$$
$$+ \eta \delta^{(m-1)}(n-1)\, v^{(m-1)}(n-1) \quad (6.14)$$

where $w^{(m)}(-1)=0$ and $|w^{(m)}(0)|<1$. The above equation can be rewritten as

$$w^{(m)}(n) = (\sum_{i=0}^{n}\alpha^i)w^{(m)}(0) + \eta\sum_{l=0}^{n-1}(\sum_{i=0}^{n-l-1}\alpha^i)\,\delta^{(m)}(l)\,v^{(m-1)}(l)$$

$$= \frac{1}{1-\alpha}[(1-\alpha^{n+1})w^{(m)}(0) + \eta\sum_{l=0}^{n-1}(1-\alpha^{n-l})\delta^{(m)}(l)\,v^{(m-1)}(l)] \qquad (6.15)$$

where $n = 0,1,2.....$

The momentum parameter in equation (6.13) is

$$0 \le \alpha < 1 \qquad (6.16)$$

For large n, $\alpha^n \ll 1$ and the weight $w^{(m)}(n)$ approaches

$$w^{(m)}(n) \approx \frac{1}{1-\alpha}[w^{(m)}(0) + \eta\sum_{l=0}^{n-1}\delta^{(m)}(l)\,v^{(m-1)}(l)]$$

$$= \frac{1}{1-\alpha}w^{(m)}(0) + \eta_{eff}\sum_{l=0}^{n-1}\delta^{(m)}(l)\,v^{(m-1)}(l) \qquad (6.17)$$

where η_{eff} is the effective learning gain defined by

$$\eta_{eff} = \frac{\eta}{1-\alpha} \qquad (6.18)$$

and $\eta_{eff} \ge \eta$ for $0\le\alpha<1$. The effective learning gain is $(1-\alpha)^{-1}$ times larger than η. The initial weights also increase by a factor of $(1-\alpha)^{-1}$. From the above analysis, the result indicates that the rate of convergence is not only affected by the learning gain, η, but also the momentum parameter, α and the initial weights, $w^{(m)}(0)$. The momentum term smooths out the high frequency variations in the weight during updating and thus allows the effective weight steps to be bigger ($\eta_{eff} > \eta$, for $\alpha\neq 0$). The effective learning gain η_{eff} as a function of η and α is shown in Figure (6.11). The result in the above is only an approximation, but it is supported by computer simulations.

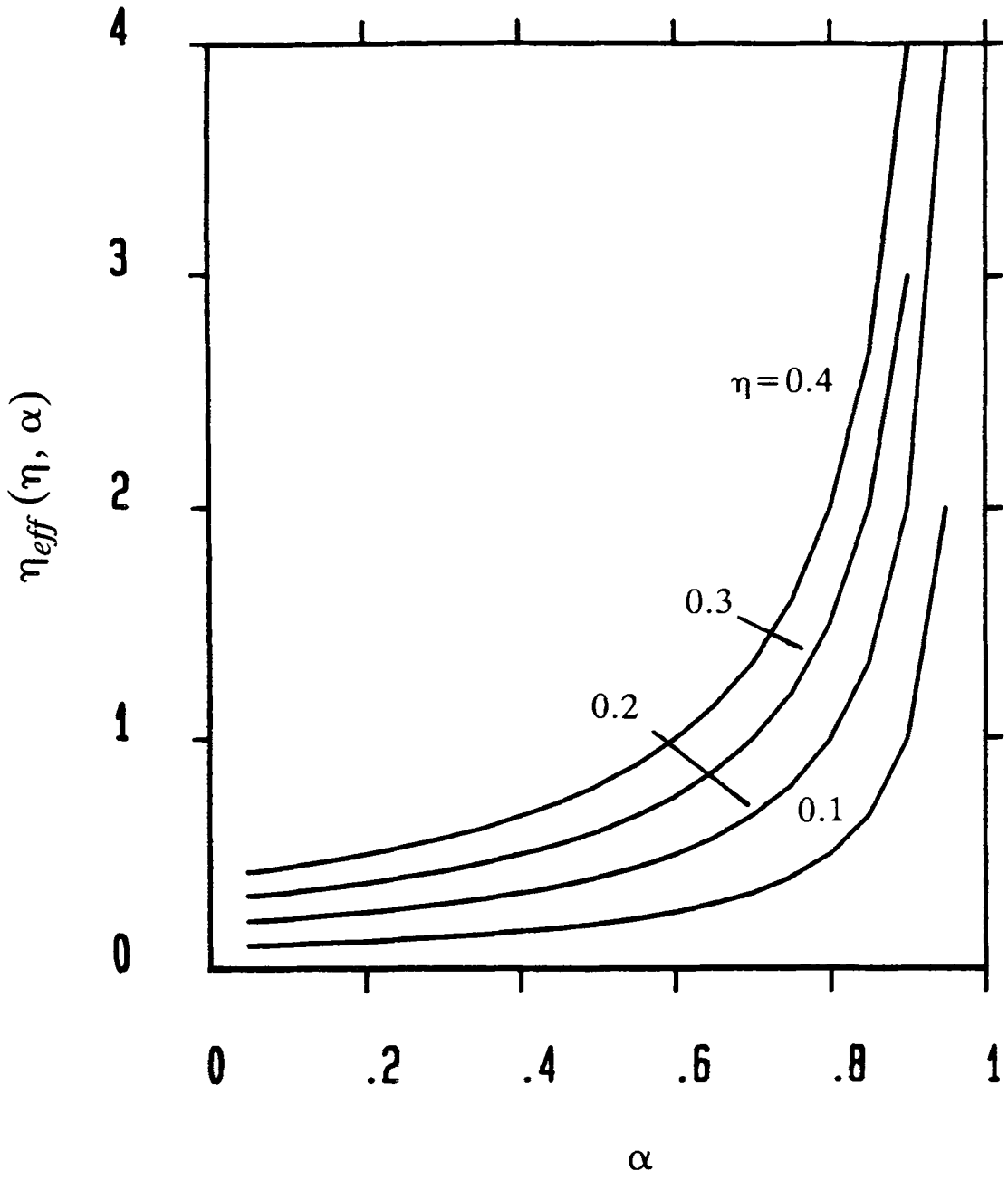


Figure (6.11) $\eta_{eff}(\eta,\alpha)$ as a function of η and α.

6.5.2 Effects of Learning Parameters and Optimization on the Performance

The performance of the multi-layer perceptron trained with the back propagation learning algorithm as affected by the choice of parameters in the learning algorithm (η, α), which is analysed in section (6.5.1), is examined in the following section using the perceptron-based equalizer scenario { (9,3,1)MLP structure with $N=5$ }. Throughout, ζ = 2.0, the channel equation in (4.14) and sigmoid function in (3.3) were chosen for performance simulation.

6.5.2.1 Convergence Characteristics

The analysis results show that the convergence time is directly proportional to the term (1 - α) and inversely proportional to the effective learning gain $\eta_{eff}(\eta, \alpha)$. Also the convergence time is dependent only on the value of $\eta_{eff}(\eta, \alpha)$, and for the same convergence time a large η requires a small α and vice versa. Figures (6.12), (6.13) and (6.14) show the convergence time as a function of $\eta_{eff}(\eta, \alpha)$. From these Figures, we can see

(1) The convergence time can be improved by a factor of $(1-\alpha)^{-1}$ with a momentum term for a given value of η.

(2) The convergence time is inversely proportional to η_{eff} and is also dependent only on the value of η_{eff}. As an example (see Figure (6.14)) both sets of parameters { η = 0.4, α = 0.0 } and { η = 0.03 , α = 0.925 } give the same values of η_{eff}, this is equal to 0.4, also the convergence time to the noise floor (- 20 dB) of both is about 235 samples.

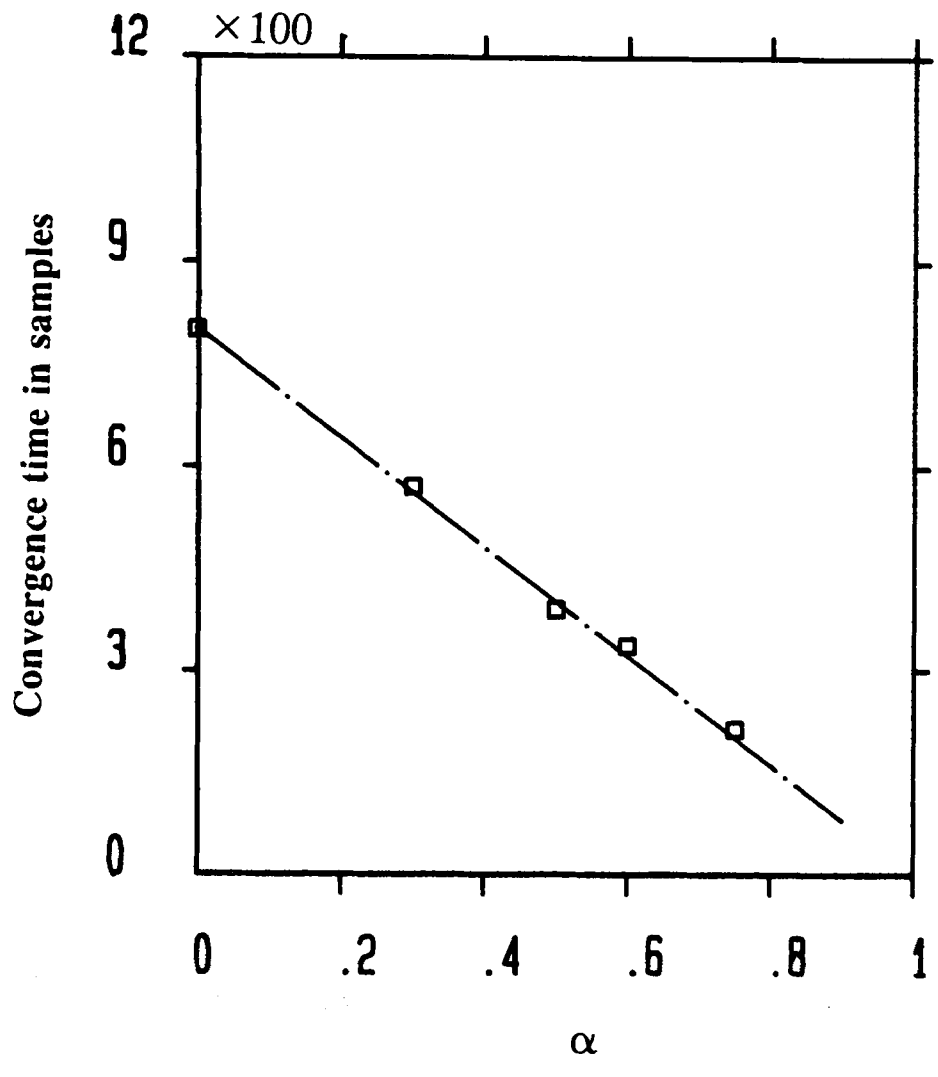


Figure (6.12) The convergence time to the noise floor (-20 dB) for a perceptron-based equalizer, { (9,3,1)MLP structure with N=5 } as a function of α, and $\eta = 0.1$ at SNR = 20 dB.

The squares " □ " show the simulation data.

The dot-dashed line shows the slope of $(1 - \alpha)$ for reference.

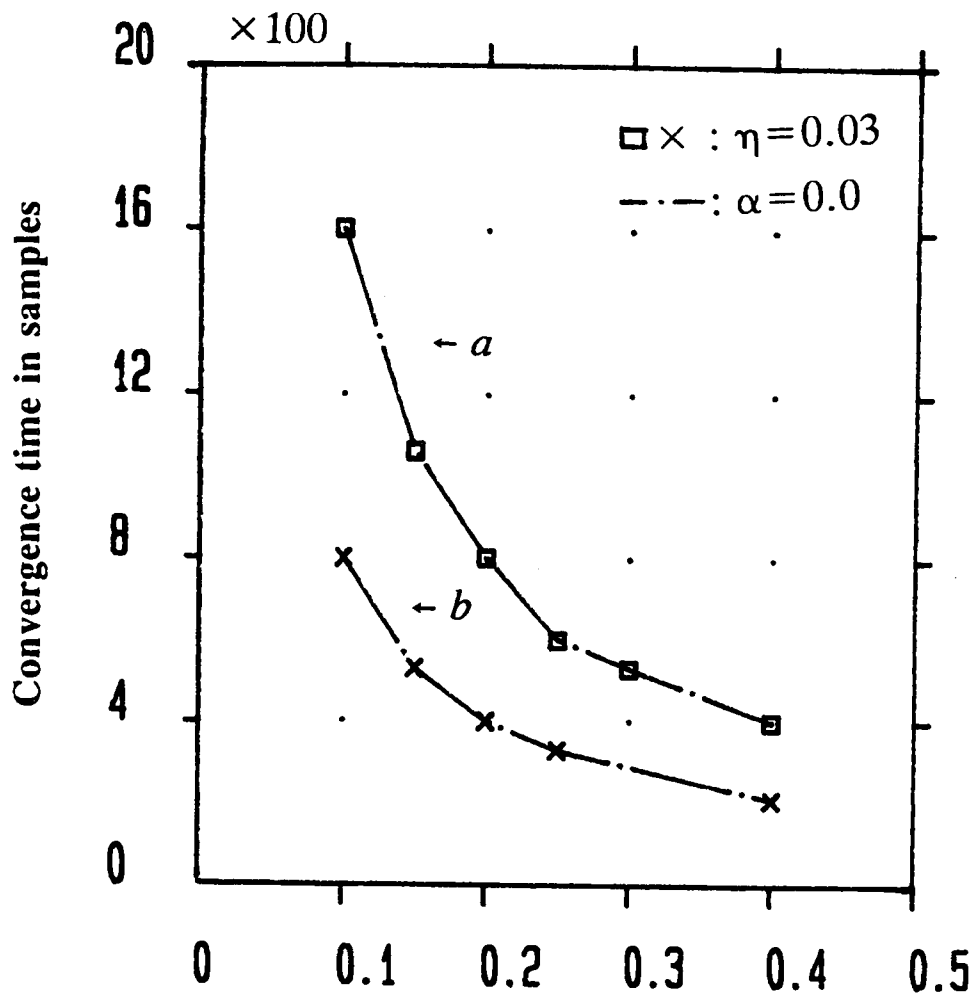
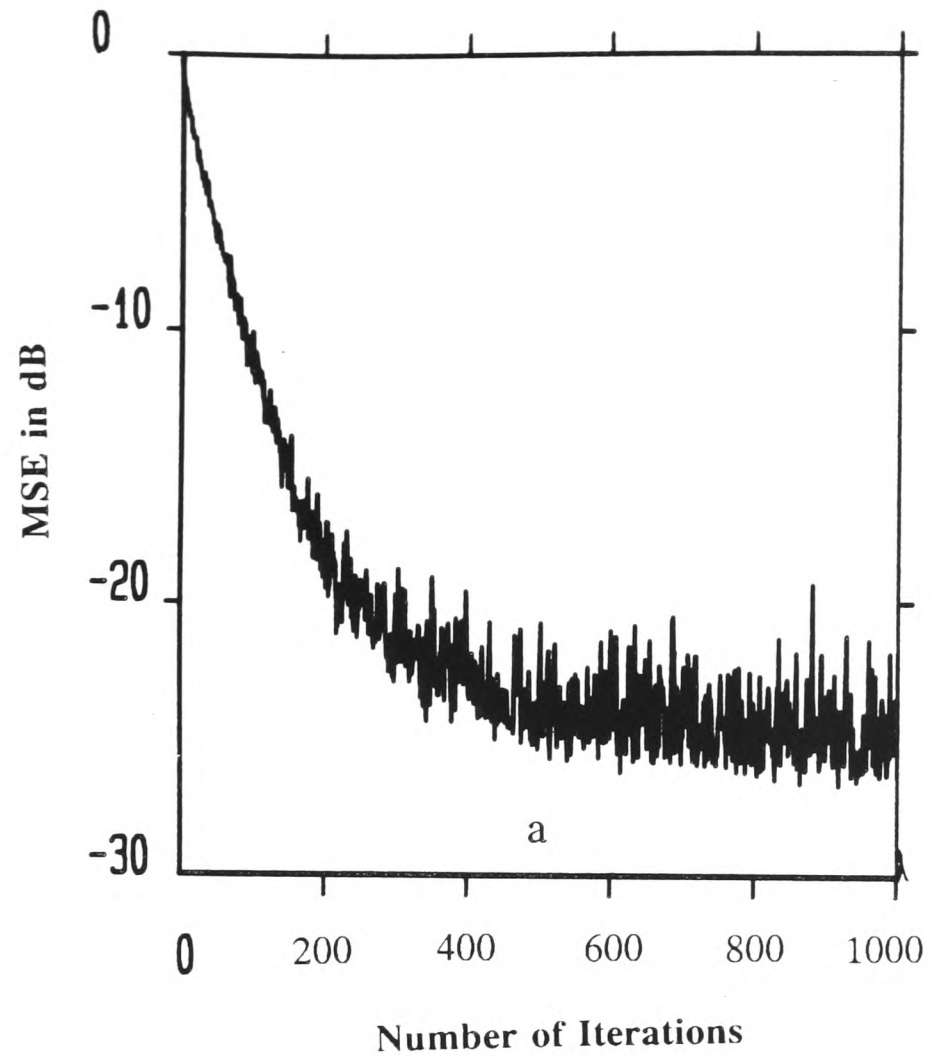


Figure (6.13) Simulation results showing the convergence rate performance for a perceptron-based equalizer, { (9,3,1)MLP structure with N=5 } as a function of $\eta_{eff}(\eta, \alpha)$ at SNR = 20 dB.

(a) The convergence time to the steady state error,

(b) The convergence time to the noise floor (-20 dB).



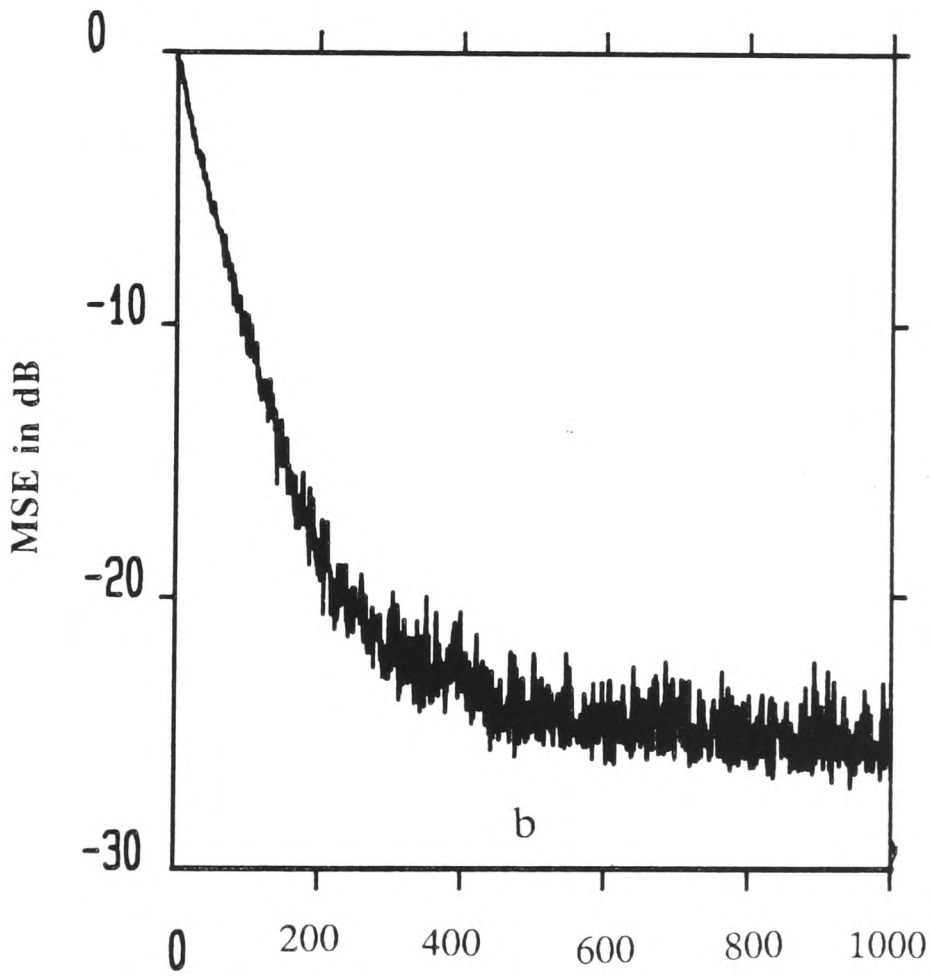


Figure (6.14) Simulation results showing relative convergence rate performance for a perceptron-based equalizer, { (9,3,1) MLP structure with N=5 } for $\eta_{eff}(\eta, \alpha) = 0.4$ at SNR = 20 dB.

(a) $\eta = 0.4$, $\alpha = 0.0$, and $\beta = 0.05$.

(b) $\eta = 0.03$, $\alpha = 0.925$, and $\beta = 0.05$.

(3) For the same value of $\eta_{eff}(\eta,\alpha)$, those with a momentum term($\alpha \neq 0$) give low variance of the mean square error (MSE). The reason is because the momentum term smooths out high frequency variations in the weight vector during weight updating process. The frequency bandwidth is determined by the momentum parameter " α ".

From the above observations the results are consistent with the analysis.

6.5.2.2 Bit Error rate performance - Decision-Directed Mode

In this section, the effect of η and α on the bit error rate performance of perceptron-based equalizer in terms of $\eta_{eff}(\eta, \alpha)$ is explored. Figure (6.15) shows the bit error rate performance as a function of $\eta_{eff}(\eta, \alpha)$ for different values of η , and Figure (6.16) shows the bit error rate performance as a function of α for different values of η at SNR = 20 dB. From these Figures, better bit error rate performance for small value of $\eta_{eff}(\eta, \alpha)$ is apparent. The performance degraded significantly for $\eta_{eff}(\eta, \alpha)$ greater than $\eta^{*}_{eff}(\eta, \alpha)$, where $\eta^{*}_{eff}(\eta, \alpha)$ is the effective learning gain corresponds to the minimum value of BER for a η in Figure (6.15). The values of α corresponding to $\eta^{*}_{eff}(\eta, \alpha)$ for η = 0.1, 0.07, and 0.05 are very similar and approximately equal to 0.5, as shown in Figure (6.16). When $\eta_{eff} < \eta^{*}_{eff}$, the bit error rate is slightly reduced as α increases, as shown in Figure (6.15). Furthermore, the bit error rate performances correspond to $\eta^{*}_{eff}(\eta, \alpha)$ for different values of η are very similar, as shown in Figure (6.15) and Figure (6.16). The above results are observed in the case of high signal to noise ratio (20 dB).

Figure (6.17) shows the bit error rate performance as a function of signal to noise ratio (SNR) for the same value of $\eta_{eff}(\eta,\alpha)$ with different values of η. In Figure

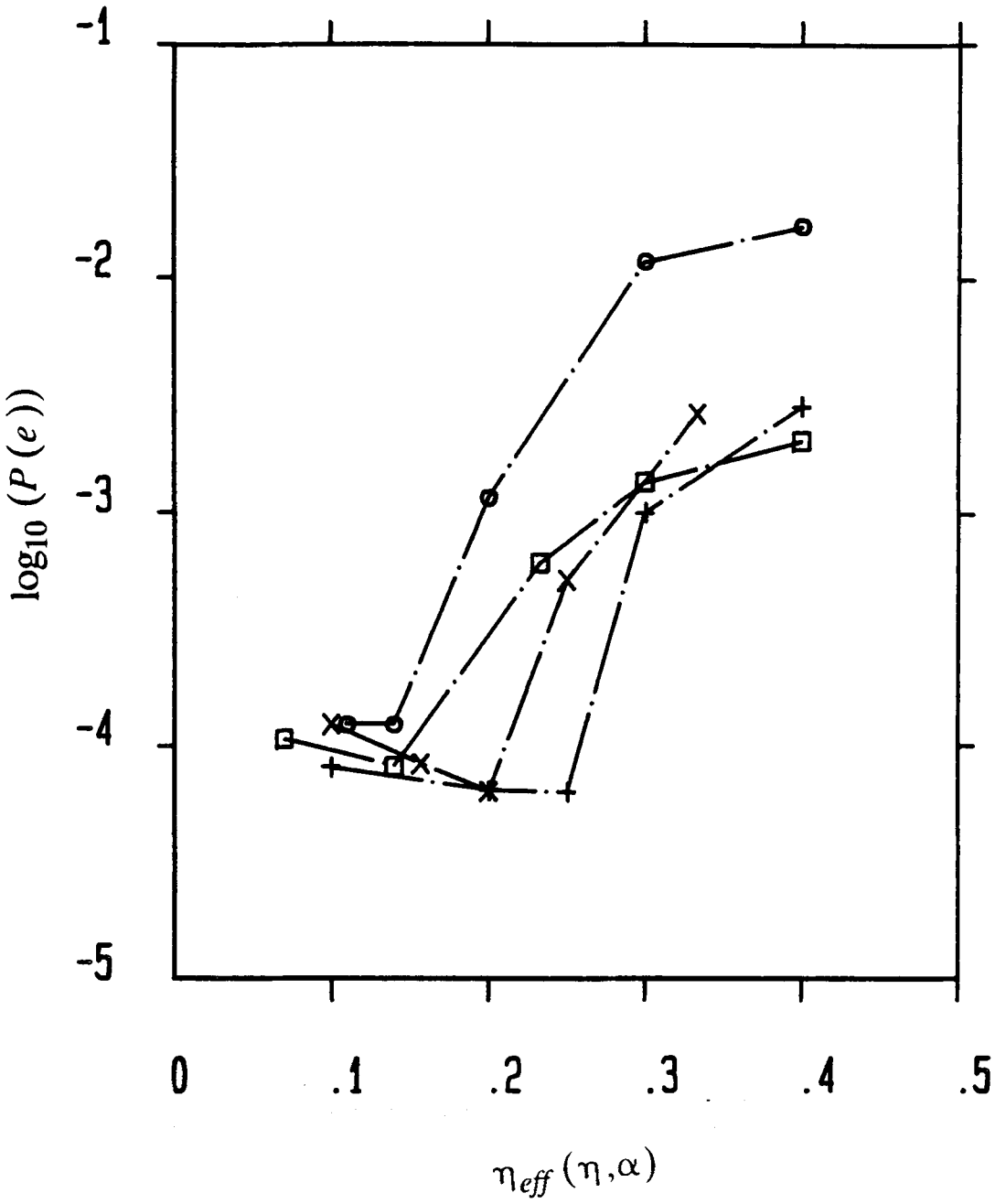


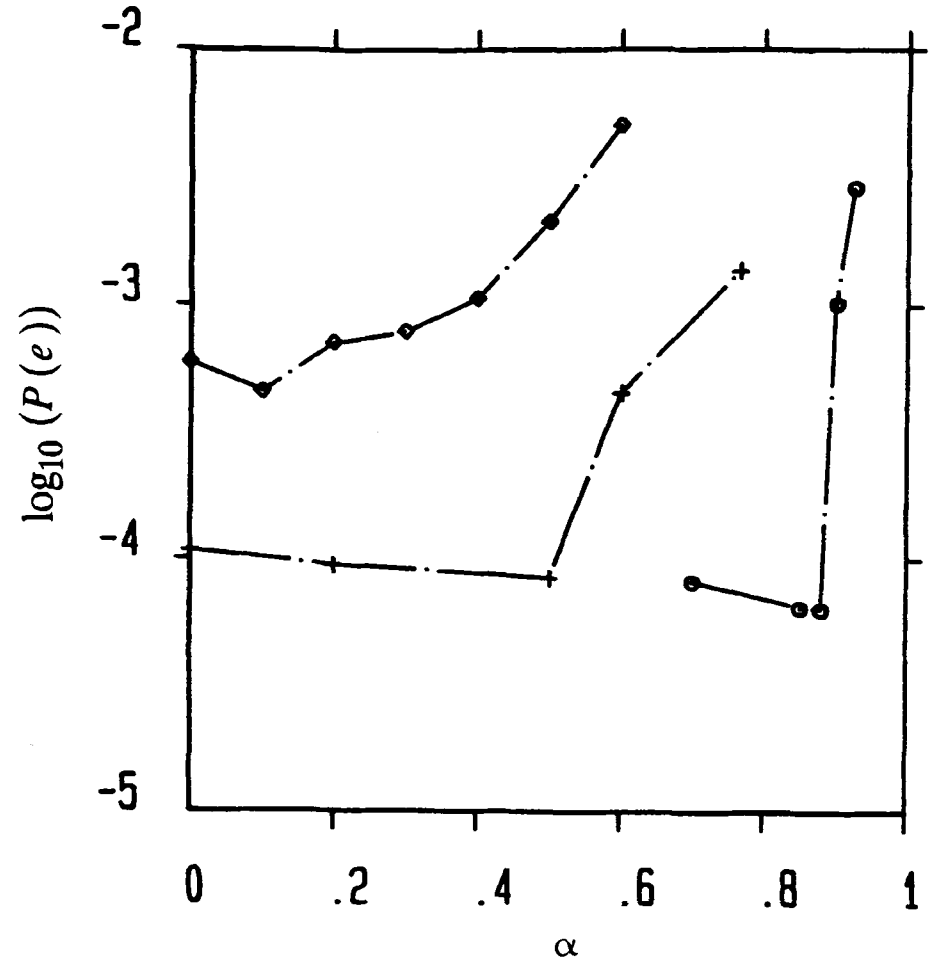
Figure (6.15) The bit error rate performance of the perceptron-based equalizer, { (9,3,1)MLP structure with N=5 } as a function of $\eta_{eff}(\eta, \alpha)$, equation (6.18), at SNR = 20 dB.

o : $\alpha = 0.0$, $\beta = 0.05$,

× : $\eta = 0.1$, $\beta = 0.05$,

□ : $\eta = 0.07$, $\beta = 0.05$,

\+ : $\eta = 0.03$, $\beta = 0.05$.



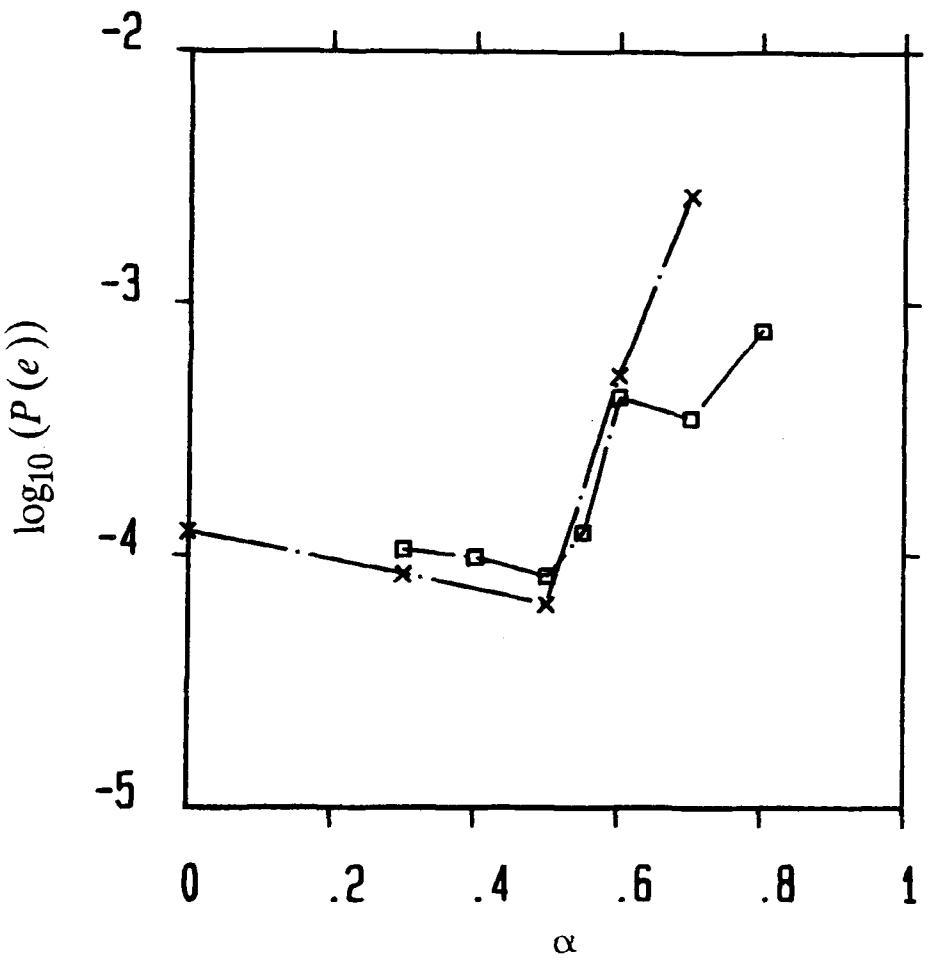


Figure (6.16) The bit error rate performance of the perceptron-based equalizer, { with (9,3,1)MLP structure with N=5 } as a function of α for different values of η at SNR = 20 dB.

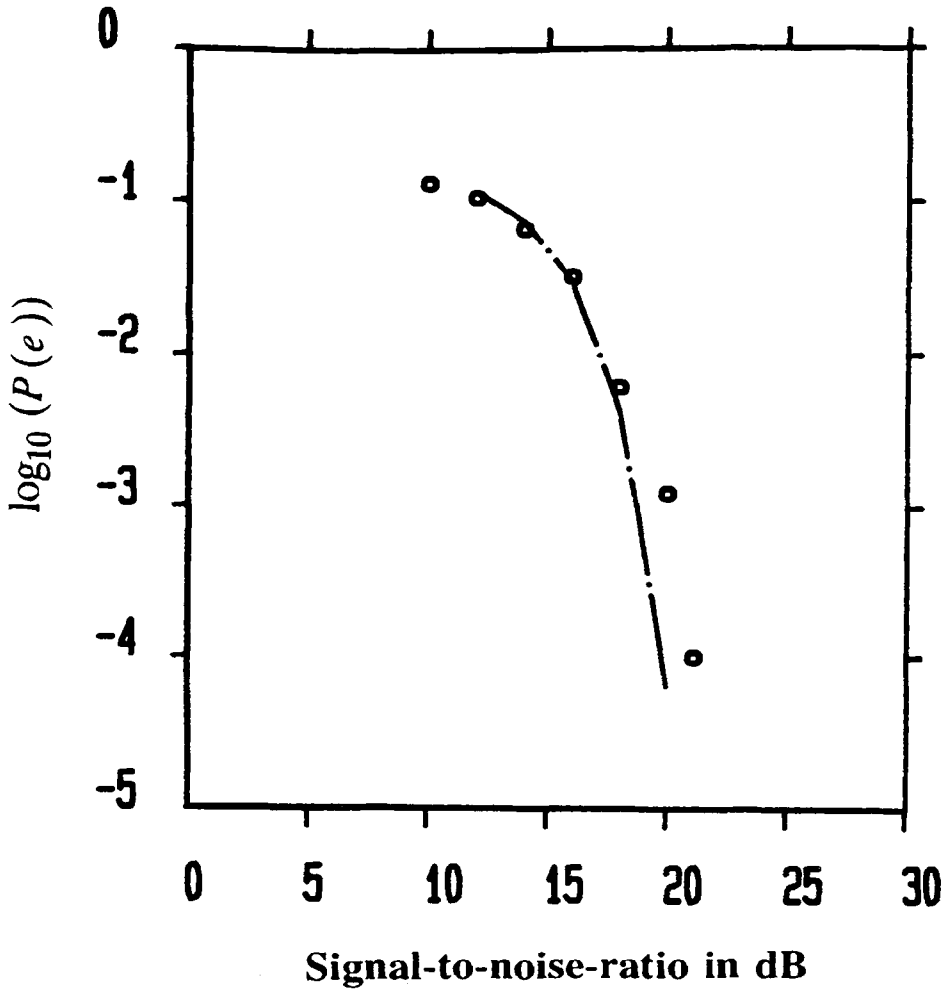
◇ : η = 0.15, β = 0.05;

\+ : η = 0.07, β = 0.05,

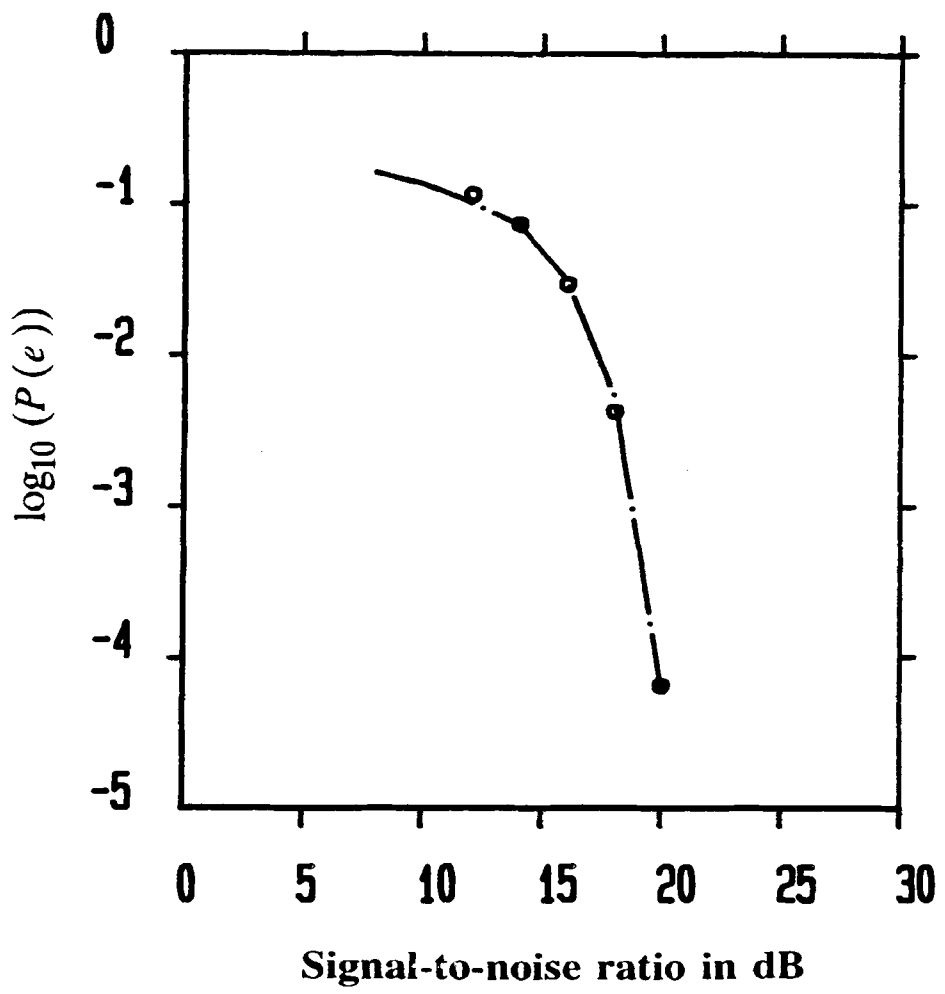
o : η = 0.03, β = 0.05;

× : η = 0.1, β = 0.05;

□ : η = 0.05, β = 0.05.



(a)



(b)

Figure (6.17) The bit error rate performance of the perceptron-based equalizer, { (9,3,1)MLP structure with N=5 } for $\eta_{eff}(\eta, \alpha) = 0.2$.

(a) $-\cdot-$: $\eta = 0.03$, $\alpha = 0.85$, $\beta = 0.05$;

o : $\eta = 0.2$, $\alpha = 0.0$, $\beta = 0.05$.

(b) $-\cdot-$: $\eta = 0.1$, $\alpha = 0.5$, and $\beta = 0.05$;

o : $\eta = 0.03$, $\alpha = 0.85$, $\beta = 0.05$.

(6.17), the value for η_{eff} is 0.2, for all three sets of learning parameters. In Figure (6.17(a)), the result indicates that the perceptron equalizer with a smaller value of η and $\alpha \neq 0$ enjoys a lower bit error rate performance. The improvement is more significant as the signal to noise ratio increases. While in low signal to noise ratio they are very similar. In Figure (6.17(b)), the $\eta_{eff}(\eta, \alpha)$ for both sets of the learning parameter corresponds to $\eta^{*}_{eff}(\eta, \alpha)$. The results shows that both sets of the learning parameter exhibit approximately the same bit error rate performance

Figure (4.10(a)) in chapter 4 shows the effect of α on the bit error rate performance as a function of signal to noise ratio. For the same learning gain(η), the bit error rate performance degraded significantly as the signal to noise ratio is decreased for those with a momentum term (α). The reason is because the effective learning gain η_{eff} is larger than η resulting in greater misadjustment at the weights. However, in the case of high signal to noise ratio the performance is slightly improved as α is increased.

From the simulation and analysis results, we see that it is safer to choose small values for both $\eta_{eff}(\eta, \alpha)$ and η, where $\eta_{eff}(\eta, \alpha) \leq \eta^{*}_{eff}(\eta, \alpha)$. This minimizes misadjustment, weight changes are smoothed and provides low bit error rate performance after convergence.

6.6 Discussion

It has been shown that as ζ (power metric) decreases, the convergence time tends to reduce roughly linearly, and a lower value of both the steady-state mean square error and the bit error rate performance can be achieved.

Choice of parameters in the learning algorithm can affect the system performance considerably. An effective learning gain has been presented that relates the learning gain

and the momentum parameter. The convergence time is dependent only on the value of the effective learning gain, and for the same convergence time a large learning gain requires a small momentum parameter and vice versa. For the same value of the effective learning gain, those with a momentum term gives low variance of the mean square error and also provide better bit error rate performance. Also for small η the same value of effective learning gain $\eta_{eff}(\eta, \alpha)$, where $\eta_{eff}(\eta, \alpha) \leq \eta^{*}_{eff}(\eta, \alpha)$ they exhibit approximately the same bit error rate performance. Furthermore, the convergence time can be improved by a factor of $(1-\alpha)^{-1}$ with a momentum term for small η.

CHAPTER 7

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

7.1 General Remarks

The subject of this thesis is the original study of the application of multi-layer perceptron architecture for channel equalization in digital communication systems. To this end both the perceptron-based equalizer and perceptron-based DFE have been considered. The factors which affect their performance have been evaluated including network topology, cost function, choice of parameters in the learning algorithm, and complexity of decision regions. Comparisons of the BER and the convergence performance of this perceptron-based equalizer with a conventional linear equalizer are provided. The subsequent paragraphs summarize the conclusions which have been drawn in this thesis.

7.2 Summary

In the multi-layer perceptron structure p neurons with a hard-limiting function in the first hidden layer form p hyperplanes in a $(\phi-1)$ dimensional space for ϕ dimensional input space. The p hyperplanes can form a maximum of $C(p,\phi)$ polyhedral sets (see equation (3.5)). The polyhedral set forms the basic building block of the decision

region for the multi-layer perceptrons. For two-layer perceptrons, the output layer can group the polyhedral sets to form the decision region including convex, non-convex, and disconnected regions. In the multi-layer perceptrons for more than one hidden layer, the operation of the neuron in the second hidden layer is to group the polyhedral sets in the first hidden layer to form intermediate decision regions, and so on. Ultimately the neuron in the final layer groups the intermediate decision regions in the layer next to it to form the output decision region. Since each of the intermediate decision regions is a finite union of polyhedral sets, so is the output decision region.

The polyhedral sets can be associated with classes. That is, the $C(p,\phi)$ polyhedral sets may be merged into F classes, where $F \leq C(p,\phi)$. Furthermore, the number of separable polyhedral sets identifies the minimum number of input training samples.

The neuron with a hard-limiting function partitions space with a hyperplane , but this is still linear in nature. The use of the sigmoid function allows different gradations of output and can make curved partitions of the space to form decision regions which have highly nonlinear decision boundaries. The complex nonlinear decision boundary can overcome the nonlinearity effects caused by the transmission channel, in case of equalization.

For the multi-layer perceptron, as the number of neurons within each hidden layer is increased, there is initially a very significant improvement in learning speed, but this improvement becomes less significant as the number of neurons further increases.

The choice of parameters in the learning algorithm can affect the system performance considerably. An effective learning gain has been presented that relates the learning

gain and the momentum parameter. The convergence time is dependent on the value of the effective learning gain, and for the same convergence time a large learning gain requires a small momentum parameter and vice versa. Furthermore, for the same value of the effective learning gain, those with a momentum term gives low variance of the mean square error and also provides better bit error rate performance. For small η the same value of effective learning gain $\eta_{eff}(\eta, \alpha)$, where $\eta_{eff}(\eta, \alpha) \leq \eta^{*}_{eff}(\eta, \alpha)$ they exhibit approximately the same bit error rate performance. The convergence time can be improved by a factor of $(1 - \alpha)^{-1}$ with a momentum term for small value of η.

The perceptron-based equalizer forms a decision region which is close to that achieved by the optimal equalizer, when the SNR is low, but deteriorates in comparison as the SNR improves. Since, if the additive noise level is very low, the perceptron will receive very few input samples which are close to the optimal decision boundary, rendering it incapable of forming the optimal decision region as it does in the high noise situation. Furthermore, when the SNR is high, the decision boundary of the perceptron-based equalizer has a piecewise linear boundary which coincides with the optimal linear equalizer decision boundary " hyperplane " in the central region. Thus, we can expect the performance of the perceptron-based equalizer to be near that of the optimal linear equalizer at high SNR, and to approach it asymptotically in the limit.

The perceptron- based equalizer offers significant reduction in the bit error rate over the conventional linear equalizer(i.e. LMS linear transversal equalizer). Because it has the ability to form highly nonlinear decision regions, in contrast with the linear equalizer which only forms linear decision regions. The linearity of the decision regions limits the performance of the conventional linear equalizer, where a nonlinearity exists

in the channel. Furthermore, the perceptron is less susceptible to the effects of high levels of additive noise relative to the conventional linear equalizer.

The conventional structure of the decision feedback equalizer (DFE) consists of a feedforward equalizer and a feedback filter, where the feedforward equalizer is linear. Basically, the conventional DFE structure is linear in nature. The linearity of the feedforward equalizer limits the performance of the DFE. A new structure for the DFE using the multi-layer perceptron structure has been presented. The back propagation learning algorithm is applied directly to the multi-layer perceptrons with a decision feedback signal vector. From comparison of simulation results it can been seen that the perceptron-based DFE provides better BER performance relative to the LMS DFE, especially in poor signal to noise ratio, also that BER performance degrades less due to decision errors and is also less sensitive to the learning gain variation.

Finally, a method for improving the convergence time and the BER performance of the perceptron-based equalizer has been presented which is based on the Minkowski-ζ power metrics. The results indicate that as ζ decreases the convergence time tends to improve roughly linearly, and a lower value of both the steady-state mean square error and the bit error rate performance can be achieved.

7.3 Further work

The study of the multi-layer perceptron-based equalizer have so far been mainly empirical, since the high dimensionality and degree of nonlinearity of the networks are extremely difficult to analyse. Empirical studies on a selected set of suitable channel data have produced some useful heuristics and rules of thumb for understanding the performance behaviour, but there is widespread agreement on the need for a more

basic theoretical understanding. This includes investigation of the convergence performance [68], examination of the effect of local minima in the system error surface [11] on a larger number of simulated channels, and sizing of the network to established the bounds on the number of neurons which are typically required in each layer. Further, the back propagation algorithm is slow in learning, and a new learning algorithm which can overcome this problem, and the problems mentioned above, is a vital topic for future research.

APPENDIX A

The k^{th} order Volterra series where the input-output relationship is given by:

$$\hat{y}_n = h_0 + \sum_{i_1=0}^{N-1} h_{i_1}^{(1)} x_{n-i_i} + \sum_{i_1=0}^{N-1}\sum_{i_2=0}^{N-1} h_{i_1 i_2}^{(2)} x_{n-i_1} x_{n-i_2} + \cdots\cdots\cdots + \sum_{i_1=0}^{N-1} \cdots\cdots\cdots \sum_{i_k}^{N-1} h_{i_1 \cdots\cdots i_k}^{(k)} x_{n-i_1} \cdots\cdots x_{n-i_k} \qquad (A.1)$$

where x_n and $\hat{y}_n$ are input and output sequences, respectively and N is the number of delays involved. The coefficient sequence $\{ h_{i_1}^{(k)} \}$ for $k=1$ and $0 \leq i_i \leq N-1$ is the impulse response of the linear part of the system. The higher order sequence $\{ h_{i_1 \cdots\cdots i_k}^{(k)} \}$ for $k \geq 2$ and $0 \leq i_k \leq N-1$ can be viewed as higher order impulse response which characterize the various orders of nonlinearity of the system. The adaptive algorithms include the LMS, and RLS linear algorithm.

REFERENCES

[1] S.U.H.Qureshi, " Adaptive Equalization ", Proc. IEEE Vol. 73, No. 9, Sept., 1985, pp. 1349-1387.

[2] C.F.N.Cowan, " Performance Comparisons of Finite Linear Adaptive Filters " , IEE Proceedings, Vol. 134, Pt. F, No. 3, June, 1987, pp. 211-216.

[3] M.Schetzen,' The Volterra and Wiener Theory of Nonlinear System ' , John Wiley, New York, 1980.

[4] M.Schetzen, " Nonlinear System Modeling Based on the Wiener Theory ", Proc. IEEE, Vol. 69, No. 12, Dec., 1981.

[5] E.Biglieri, " Theory of Volterra Processors and Some Applications " , Proc. IEEE Int. Conf. ASSP, 1982, pp. 294-297.

[6] V.J. Mathews and J.Lee, " A Fast Recursive Least-Squares Second Order Volterra Filter ", Proc. IEEE Int. Conf. ASSP, 1988, pp. 1383-1386.

[7] R.P.Lippmann, " An Introduction to Computing With Neural Network " IEEE ASSP Magazine, vol. 4, pp. 4-22, April, 1987.

[8] T. Kohonen," An Introduction to Neural Computing ", Neural Networks ,Vol. 1 No. 1, pp. 3-61, 1988.

[9] C.F.N.Cowan and P.F.Adams, " Non-linear System Modelling : Concept and Application ", Proc. IEEE Int. Conf. ASSP, 1984, pp.45.6.1-45.6.4.

[10] J. Chen and J. Vandewalle, " Study of Adaptive Nonlinear Echo Canceller With Volterra Expansion ", Proc. IEEE Int. Conf. ASSP, 1989, pp. 1376-1379.

[11] S.Benedetto, E.Biglieri, and V.Castellani,' Digital Transmission Theory ', Prentice-Hall, Inc., 1987.

[12] S.Benedetto and E. Biglieri, "Nonlinear Equalization of digital Satellite Channels," Ninth AIAA Conference on Communication Satellite, San Diego, CA, March 7-11, 1982.

[13] A.Lapedes, and R.Farber, " Nonlinear Signal Processing Using Neural Networks: Prediction, and System Modelling ", Preprint, LA-UR87-2662, 1987 (Los Alamons National Lab.).

[14] J.D.Gibson, ' Principles of Digital and Analog Communications ', Chapter 9, pp. 252-283, Macmillan Publishing Company, 1989.

[15] J.G. Proakis, ' Digital Communications ', Chapter 6, pp. 519-701, McGraw-Hill Book Company, 1989.

[16] A.A.Giordano, and F.M. Hsu,' Least Square Estimation with Applications to Digital Signal Processing ', chapter 6, John Wiley & Sons, 1985.

[17] S.Haykin,' Adaptive Filter Theory ', Prentice-Hall, Inc., 1986.

[18] M.D.Srinath, and P.K.Rajasekaran, ' An Introduction to Statistical Signal Processing with Applications ', Chapter 3, New York, Wilsey, 1979.

[19] J.L.Melsa, and David L.Cohn,‘ Decision and Estimation Theory ’, Chapters 3-4, New York Mc-Graw-Hill, 1978.

[20] H.L.Van Trees, ‘ Detection, Estimation, and Modulation Theory ’, Part 1, New York: Wiley, 1968.

[21] F.A.Graybill,‘ Matrices with Applications Statistics ’, Wadsworth, Inc., 1983.

[22] L.W.Couch II,‘ Digital and Analog Communication Systems ’, Chapter 8, Macmillan Publishing Co., Inc., 1983.

[23] W.S.McCulloch, and W.Pitts, " A Logical Calculus of the Ideals Imminent in Nervous Activity ", Bulletin of Mathematical Biophysics, 5, pp. 115-133, 1943.

[24] D.O.Hebb, ‘ The Organization of Behavior ’, John Wiley & Sons, New York , 1949.

[25] R.Rosenblatt,‘ Principles of Neurodynamics ’ , New York, Spartan Books, 1959.

[26] B.Widrow, and M.E.Hoff, " Adaptive Switching Circuits ", 1960 IRE Wescon Conv. Record, Part 4, pp. 96-104, August, 1960.

[27] M.Minsky, and S.Papert,‘ Perceptron: An Introduction to Computational Geometrical ’, MIT Press, 1969 (Expanded Edition, 1988).

[28] J.J.Hopfield, " Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons ", Proc. Natl. Acad. Sci. USA, Vol. 81, pp. 3088-3092, May, 1984.

[29]J.J.Hopfield, " Neuron Networks and Physical Systems with Emergent Collective Computational Abilities ", Proc. Natl. Acad. Sci. USA, Vol. 79, pp. 2554-2558, April, 1982.

[30]J.J.Hopfield,and D.W.Tank, " Computing with Neural Circuits : A Model ", Science, Vol. 233, pp. 625-633, August, 1986.

[31] D.E.Rumelhart, G.E.Hinton and R.J.Williams, "Learning Internal Representations by Error Propagation ", in Parallel Distributed Processing : Explorations in the Microstructure of Cognition, (D.E.Rumelhart, and J. L.McCelland) Chapter 8, Vol. 1, MIT. Press, Cambridge, MA (1986).

[32] T.Sejnowski et al., " Net Talk: A Parallel Network that Learns to Read Aloud ", Johns Hopkins Univ. Preprint (1986).

[33] J.A.Feldman, and D.H.Ballard, " Connectionist Models and Their Properties ", Cognitive Science, Vol. 6, pp. 205-254, 1982.

[34] S.Grossberg, ' The Adaptive Brain 1: Cognitive ', ' Learning Brain 11: Vision, Speech, Language, and Motor Control ', Elsevier/North-Holland, Amsterdam, 1986.

[35] S.Grossberg,' Neural Networks and Natural Intelligence ' , MIT Press, 1988.

[36] D.J.Burr, "A Neural Network Digit Recognizer", Proc. IEEE Int. Conf. System, Man, and Cybernetics, 1986, pp. 1621-1625.

[37] D.J.Burr, " Experiments on Neural Network Recognition of Spoken and Written Text ", IEEE Trans. ASSP, Vol. 36, No. 7, July, 1988, pp. 1162-1168.

[38] Y.Le Cun et al., " Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning ", IEEE Com. Magazine , pp.41-46, Nov., 1989.

[39] R.P.Lippmann, " Pattern Classification Using Neural Networks ", IEEE Com. Magazine, pp. 47-64, Nov., 1989.

[40] B.P.Yuhas, " Integration of Acoustic and Visual Speech Signals Using Neural Networks ", IEEE Com. Magazine, pp. 65-71, Nov., 1989.

[41] T.X.Brown, " Neural Networks for Switching ", IEEE Com. Magazine, pp.72-81, Nov., 1989.

[42] J.C.Lupo, " Defense Applications of Neural Networks ", IEEE Com. Magazine, pp. 82-87, Nov., 1989.

[43] B.Widrow, R.G.Winter, and R.A.Baxter, " Layered Neural Nets for Pattern Recognition ", IEEE Trans. on ASSP, Vol. 36, No. 7, July, 1988.

[44] W.Y.Huang and R.P.Lippmann, " Neural Net and Traditional Classifiers ", AIP Conf. Proc. Neural Information Processing Systems, Denver, Co, 1987.

[45] Y.H.Poa,' Adaptive Pattern Recognition and Neural Networks ', Addison-Wesley Publishing, 1989.

[46] A.Lapedes, and R.Farber, " How Neural Nets Works ", AIP Conf. Proc. Neural Information Processing Systems, Denver, Co, 1987.

[47] S.R.Lay, ' Convex Sets and their Applications ', John Wiley & Sons, 1982.

[48] L.Schlaffi(1814-1895)," Gesammelte Mathematische Abhandlungen ", vol. 1, Birkhauser, Basel, 1950. pp. 209-212.

[49] J.Makhoul, R.Schwartz, and A.E.Jaroudi, " Classifications Capabilities of Two layer Neural Nets ", Proc. IEEE Int. Conf. ASSP, pp. 635-638, May, 1989.

[50] G. Mirchandani, and W.Cao, " On Hidden Nodes for Neural Nets ", IEEE Trans. Circuit and Systems, Vol. 36 No. 5, pp. 661-664, May 1989.

[51] G.J.Gibson, S.Siu, and C.F.N.Cowan, " Multilayer Perceptron Structures Applied to Adaptive Equalizers for Data Communications " , Proc. IEEE Int. Conf. ASSP, pp. 1183-1186, May, 1989.

[52] G.J.Gibson, S.Siu, and C.F.N.Cowan, " Application of Multilayer Perceptrons as Adaptive channel Equalizers ", pp. 323-328, Preprints of IFAC Symposium, Glasgow, U.K. 19-21 April, 1989.

[53] A.Wieland and R.Leighton, " Geometric Analysis of Neural Network Capabilities ", Proc. IEEE 1 st. Int. Conf. Neural Network, June, 1987.

[54] B.Irie, and S.Miyake, " Capabilities of Three-Layered Perceptrons " , IEEE Int. Conf. Neural Networks, San Diego, 1988.

[55] G.G.Lorentz, " The 13th Problem of Hibert ", in F.E.Browder (Ed.), , Mathematical Society, Providence, R.I., 1976.

[56] W.Y.Huang and R.P.Lippmann, " Comparisons Between Conventional and Neural Network Classifiers ", Proc. IEEE 1 st. Int. Conf. on Neural Network, June, 1887.

[57] F.J.Smieja, and G.D.Richards, " An Investigation of the Performance of a Neural Networks ", Edinburgh Preprint 87/418, Dept. of Physis, University of Edinburgh.

[58] A.P.Clark,' Equalizers for digital modems ', Pentch Press, London, 1985.

[59] D.L.Duttweiler, J.E.Mazo, and D.G.Messerschmitt, " An Upper Bound on the Error Probability in Decision-Feedback Equalization ", IEEE Trans. Inforamtion Theory, Vol. IT-20, pp. 490-497, July, 1974.

[60] J.J.O'Relly et al., " Error Propagation in Decision Feedback Receivers ", IEE Proc. Vol. 132, Pt. F, No. 7, pp. 561-566, Dec., 1985.

[61] A.M.de Oliveira Durate et al., " Simplified Technique for Bounding Error Statistics for DFB Receivers ", IEE Proc. Vol. 132, Pt. F, No. 7, Dec., 1985.

[62] S.Siu, G.J.Gibson, and C.F.N.Cowan, " Decision Feedback Equalization using Neural Network Structures ", Proc. IEE Int. Conf. on Neural Networks, Oct. 16-18, 1989, London, U.K.

[63] T.Kailath, ' Lectures on Wiener and Kalman Filtering ', Dept.of Elect. Eng., Stanford University, 1985.

[64] C.F.N.Cowan, and P.M.Grant, ' Adaptive Filter ', Pentice-Hall Inc., Englewood Cliffs, New Jersey, 1985.

[65] S.J.Hanson and D.J.Burr, " Minkowski- ζ Back Propagation: Learning in Connectionist Models with Non-Euclidian Error Signals ", AIP Conf. Proc. Neural Information Processing Systems, Denver, CO 1987.

[66] B.Widrow, and S.D.Stearns, ' Adaptive Signal Processing ', Prentice-Hall, New Jersey, 1985.

[67] R. Yarlagadda, " Fast Algorithms for l_p Deconvolution " , IEEE Tans. ASSP, Vol. 33, No. 1, Febr. 1985, pp. 174-182.

[68] G.Teasuro, Y. He, and S.Ahmad, "Asymptotic Convergence of Backpropagation", Neural Computation 1, 1989, pp.382-391.

RELEVANT PUBLICATIONS

[1]* S.Siu, G.J.Gibson, and C.F.N.Cowan, "Decision Feedback Equalization using Neural Network Structures and Performance Comparison with the Standard Architecture", IEE Proceedings-I Communication, Speech and Vision, vol. 137, No. 4, pp.221-225, August, 1990.

[2] S.Siu, G.J.Gibson, and C.F.N.Cowan, "Decision Feedback Equalization using Neural Network Structures", Proc. IEE Int. Conf. on Neural Networks, pp.124-128, Oct., 16-18, 1989, London, U.K.

[3] G.J.Gibson, S.Siu, and C.F.N.Cowan, "The Application of Nonlinear Structures to the Reconstruction of Binary Signals," accepted for publication in IEEE transactions on Acoustics, Speech, and Signal Processing.

[4] G.J.Gibson, S.Siu, and C.F.N.Cowan, "Multi-layer Perceptron structures Applied to Adaptive Equalizers for Data Communication," Proc. IEEE Int. Conf. ASSP, pp.1183-1186, May, 1989.

[5] G.J.Gibson, S.Siu, and C.F.N.Cowan, "Application of Multi-layer Perceptrons as Adaptive Channel Equalizers," Adaptive Systems in Control and Signal processing 1989. Selected papers from the IFAC Symposium , Glasgow, U.K., 19-21 April 1989

*Reprinted at back of thesis

(Oxford, U.K. : Pergamon), pp.573-578.

[6] G.J.Gibson, S.Siu, S.Chen, C.F.N.Cowan, and P.M.Grant, 'The Application of Nonlinear Architectures to Adaptive Channel Equalization," Proc. IEEE Int. Conf. On Communication, pp.649-653, April 16-19, 1990.

[7] C.F.N.Cowan, G.J. Gibson and S.Siu; " Data equalisation using highly non-linear adaptive architectures", Proc. SPIE Conf. on Advanced algorithms and architectures for signal processing IV', San Diego, pp. 34-43, Aug., 1989.

Decision feedback equalisation using neural network structures and performance comparison with standard architecture

S. Siu
G.J. Gibson
C.F.N. Cowan

Indexing terms: Distribution networks, Feedback

Abstract: The paper describes a new approach for a decision feedback equaliser using the multilayer perceptron structure for equalisation in digital communications systems. Results indicate that the perceptron based decision feedback equaliser provides better bit error rate performance relative to the least mean square decision feedback equaliser, especially in high noise conditions, also that bit error rate performance degrades less owing to decision errors and is also less sensitive to gain variation.

1 Introduction

Decision feedback equalisation is a technique used in digital communications systems (see Fig. 1) to equalise

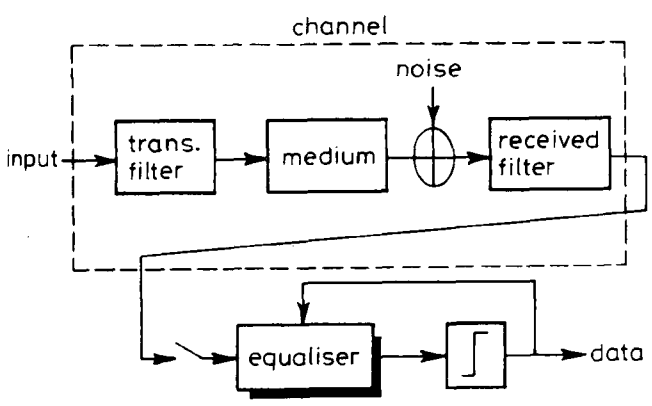


Fig. 1 *Baseband data transmission system*

the channel to remove that part of the intersymbol interference (ISI) caused by the previous data decisions. The advantage of the decision feedback equaliser is that ISI is eliminated without enhancement of noise by using past decisions to subtract out a portion of the ISI in addition to the normal feedforward filter; a disadvantage is that decision errors tend to propagate because they result in residual ISI and a reduced margin against noise at future decisions [1].

The conventional structure of the decision feedback equaliser (DFE) uses linear algorithms such as LMS (least mean square) or RLS (recursive least square) and consists of a feedforward filter and a feedback filter, as shown in Fig. 2, where the feedforward filter is a linear equaliser. The decision regions of a linear equaliser are always delimited by hyperplanes. The linearity of these decision boundaries limit the performance of the system.

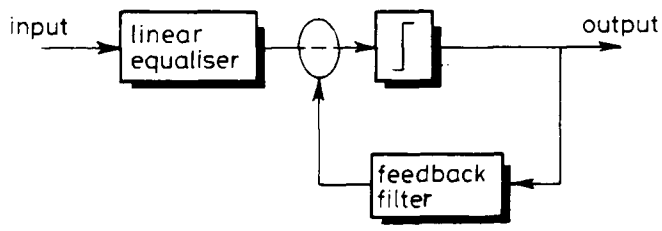


Fig. 2 *Decision feedback equaliser structure*

Artificial neural networks [2] are systems which use nonlinear computational elements to model neural behaviour based on our present understanding of the biological nervous system. A neural network may be simply considered as a nonlinear mapping between input and output. Rumelhart *et al.* (1986) [3] proposed a back propagation learning algorithm enabling multilayer perception networks [3], [4], [5] to learn more sophisticated tasks than before. The network uses a layered feedforward structure with input, output and hidden layer(s). The hidden layers provide the capability by use of the nonlinear sigmoid function, to create intricately curved partitioning of the signal space to produce nonlinear decision boundaries [6, 7].

2 Multilayer perceptrons: Architecture

The basic element of the multilayer perceptron is the neuron, which is depicted in Fig. 3. Each neuron has primarily local connections and is characterised by a set of

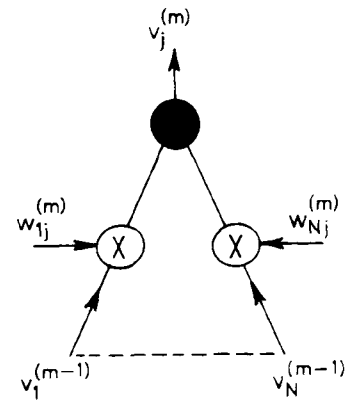


Fig. 3 *jth Neuron in mth layer*

real weights $[w_{1j}, \ldots, w_{Nj}]$ applied to the previous layer to which it is connected and a real threshold level I_j. The jth neuron in the mth layer accepts inputs $V^{(m-1)} \in R^N$ from the $(m-1)$th layer and returns a scalar $v_j^{(m)} \in R$

Paper 7358I (E7, E8), first received 13th September 1989 and in revised form 24th February 1990

The authors are at the Department of Electrical Engineering, University of Edinburgh, Mayfield Road, Edinburgh EH9 3JL, United Kingdom

given by

$$v_j^{(m)} = f_j\left(\sum_{i=1}^{N} w_{ij}^{(m)} v_i^{(m-1)} + I_j^{(m)}\right) \quad (1)$$

The output value $v_j^{(m)}$ serves as input to the $(m+1)$th layer to which the neuron is connected.

The nonlinearity commonly used in the perceptron is of the sigmoid type:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2)$$

where $f(x)$ lies in the interval $[-1, 1]$ as shown in Fig. 4. The neurons store knowledge or information in the

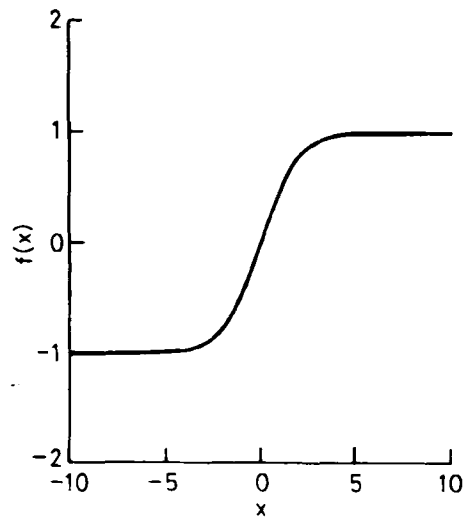


Fig. 4 *Activation function*

weights $\{w_{ij}\}$ and the weights are modified through experience or training.

Earlier work on single layer perceptrons was limited owing to the fact that only linear decision boundaries could be formed in the signal space [2]. However, later developments [3–6, 9] showed how multiple layers could be used to form much more complex (nonlinear) decision boundaries. A multilayer perceptron (MLP) consists of several hidden layers of neurons which are capable of performing complex, nonlinear mappings between the input and the output layer. The hidden layers provide the capability to use the nonlinear sigmoid's ability to create intricately curved partitions of space. In general, all neurons in a layer are fully interconnected to neurons in adjacents layers, but there is no connection within a layer, and normally no connections bridging layers, as shown in Fig. 5. Data information is recoded into the

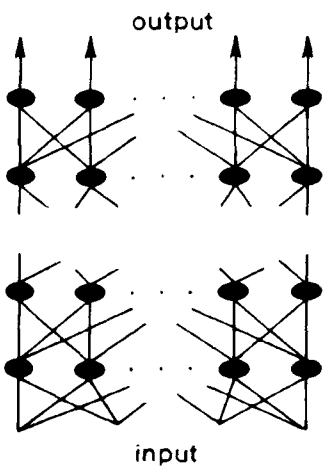


Fig. 5 *Multilayer perceptron architecture*

hidden layer(s) and the output is generated by combinational operations on the final hidden layer.

3 Perceptron-based decision feedback equaliser

A three-layer preceptron based decision feedback equaliser structure, as shown in Fig. 6, consists of a feedforward filter and a feedback filter. The input to the feedforward filter is the sequence of noisy received signal samples $\{y_n\}$. The input to the feedback filter is the

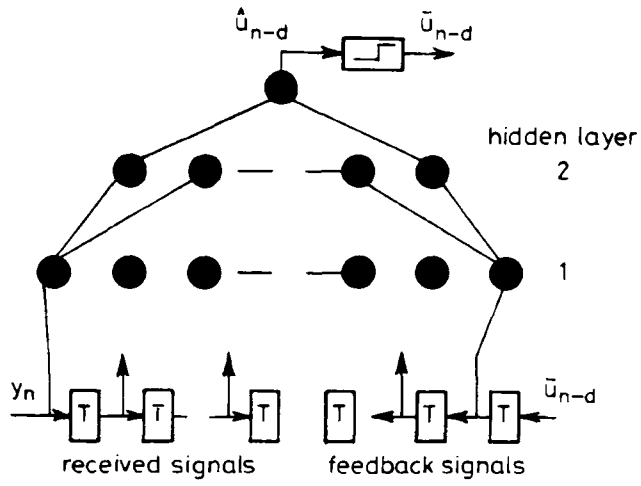


Fig. 6. *Multilayer perceptron decision feedback equaliser*

output symbol decision sequence from a nonlinear symbol detector (quantiser) $\{\tilde{u}_{n-d}\}$.

The three-layer perceptron (two hidden layers, and an output layer) is sufficient for the nonlinear DFE structure, because a three-layer perceptron can generate arbitrarily complex, nonlinear decision regions [7].

At time n, the input $N \times 1$ received signal vector

$$\Gamma(n)^T = [y_n, y_{n-1}, \ldots, y_{n-N+1}] \quad (3)$$

and the decision $l \times 1$ signal vector

$$[\tilde{u}_{n-d-1}, \tilde{u}_{n-d-2}, \ldots, \tilde{u}_{n-d-l}] \quad (4)$$

are in the feedforward filter and feedback filter of the decision feedback equaliser, respectively, where d is a delay parameter. The decision $\tilde{u}_{n-d}$ is formed by quantising the estimate $\hat{u}_{n-d}$ in the output layer to the nearest information symbol.

The signals at the input layer of the decision feedback equaliser can be represented by a $(N+l) \times 1$ vector as

$$V^{(0)} = [y_n, y_{n-1}, \ldots, y_{n-N+1}; \tilde{u}_{n-d-1}, \ldots, \tilde{u}_{n-d-l}]^T \quad (5)$$

The $N_1 \times 1$ vector in the output of hidden layer 1 is

$$V^{(1)} = [v_1^{(1)}, v_2^{(1)}, \ldots, v_j^{(1)}, \ldots, v_{N_1}^{(1)}]^T \quad (6)$$

where

$$v_j^{(1)} = f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{u}_{n-d-p} + I_j^{(1)}\right)$$

$$j = 1, 2, \ldots, N_1 \quad (7)$$

Where b denotes the feedback tap weight.

The $N_2 \times 1$ vector in the output of hidden layer 2 is

$$V^{(2)} = [v_1^{(2)}, v_2^{(2)}, \ldots, v_k^{(2)}, \ldots, v_{N_2}^{(2)}]^T \quad (8)$$

where

$$v_k^{(2)} = f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} v_j^{(1)} + I_k^{(2)}\right) \quad k = 1, 2, \ldots, N_2 \quad (9)$$

The final output is

$$v_0^{(3)} = \hat{u}_{n-d} = f_0\left(\sum_{k=1}^{N_2} w_{k0}^{(3)} v_k^{(2)} + I_0^{(3)}\right) \quad (10)$$

Where $\hat{u}_{n-d}$ is the estimated signal at time n. Substituting eqns. 7 and 9 into eqn. 10, yields

$$\hat{u}_{n-d} = f_0\left(\sum_{k=1}^{N_2} w_{k0}^{(3)} f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{u}_{n-d-p} + I_j^{(1)}\right) + I_k^{(2)}\right) + I_0^{(3)}\right) \quad (11)$$

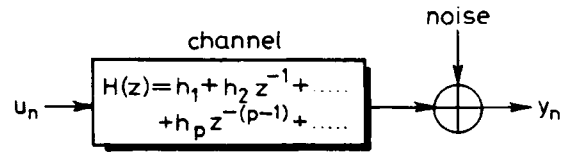
The nonlinear detector can be modelled as a threshold function $g(x)$ and is defined as

$$g(\hat{u}_{n-d}) = \tilde{u}_{n-d} = \begin{cases} 1 & \text{if } \hat{u}_{n-d} \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (12)$$

The ws (weights) and Is (threshold levels) in eqn. 11 are values specified by the training algorithm, so that after training is finished the equaliser will self-adapt to changes in channel characteristics occurring during transmission (decison directed mode).

4 Eliminating intersymbol interference: decision feedback signal

The output $v_j^{(1)}$, of the jth neuron in layer one can be expressed in terms of $\{g_p\}$, the feedback tap weights $\{w_{pj}^{b(1)}\}$ and the transmitted signal $\{u_n\}(u_n \in (1, -1))$ as shown in Fig. 7. Note that $\{g_p\}$ is the convolution of the



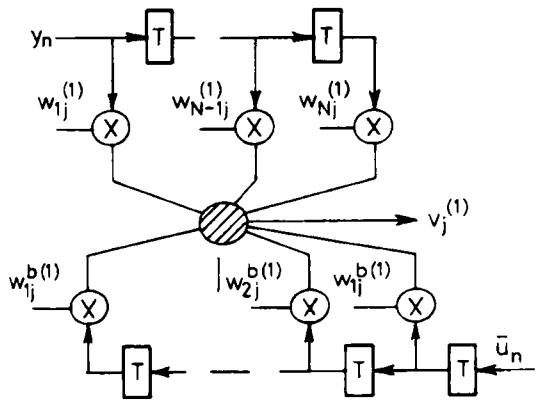


Fig. 7 *jth neuron with feedback signals in first layer*

channel impulse response $\{h_p\}$ and the weights $\{w_{ij}^{(1)}\}$. Thus

$$v_j^{(1)} = f_j\left(\sum_p u_{n-p} g_p + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{u}_{n-p} + \eta_n + I_j\right) \quad (13)$$

where η_n is zero-mean Gaussian noise.

The above equation can be written as

$$v_j^{(1)} = f_j\left(u_n g_0 + \sum_{p=1}^{l} (g_p + w_{pj}^{b(1)}) \tilde{u}_{n-p} + \sum_{p=1}^{l} (u_{n-p} - \tilde{u}_{n-p}) g_p + \sum_{\substack{p<0 \\ p>l}} u_{n-p} g_p + \eta_n + I_j\right) \quad (14)$$

If we select

$$g_p = -w_{pj}^{b(1)} \quad p = 1, 2, \ldots, l \quad (15)$$

and the probability of error is very small, we may assume that the last l symbols have been received correctly, i.e.

$$\tilde{u}_{n-p} = u_{n-p} \quad p = 1, 2, \ldots, l \quad (16)$$

Then eqn. 14, can be simplified as

$$v_j^{(1)} = f_j\left(u_n g_0 + \sum_{\substack{p<0 \\ p>l}} u_{n-p} g_p + \eta_n + I_j\right) \quad (17)$$

All ISI from past symbols ($1 \leq p \leq l$) is eliminated without altering the useful signal term $u_n g_0$ or enchancing the noise component η_n. The $\sum_{p<0,\, p>l} u_n g_p$ residual ISI term will be reduced as the signal is passed forward. If an incorrect decision is made by the detector, e.g. $\tilde{u}_{n-p} = -u_{n-p}$, the decision errors tend to propagate because they result in residual intersymbol interference and a reduced margin against noise at future decisions.

For an equaliser with (P) taps and a channel response that spans (L) symbols, the number of symbols involved in the intersymbol interference is $(P + L - 2)$. The number of taps(l) that are needed in the feedback section to eliminate all the ISI from previously detected symbols, provided that previous decisions are correct is

$$l = P + L - 2 \quad (18)$$

5 Learning algorithm

An iterative learning algorithm, called back propagation was suggested by Rumelhart *et al.* [3]. In back propagation, the output value is compared with the desired output, resulting in an error signal. The error signal is fed back through the network and weights are adjusted to minimise this error.

The increments used in updating the weights, Δw_{ij} and threshold levels, ΔI_j of the mth layer can be accomplished by the following rules:

$$\Delta w_{ij}^{(m)}(n+1) = \eta \delta_j^{(m)} v_j^{(m-1)} + \alpha \Delta w_{ij}^{(m)}(n) \quad (19)$$

and

$$\Delta I_j^{(m)}(n+1) = \beta \delta_j^{(m)} \quad (20)$$

where η is the learning gain, α is the momentum parameter, β is the threshold level adaptation gain, and layer $m \in [1, 2, \ldots, M]$.

The error signal $\delta_j^{(m)}$ for layer m is calculated starting from the output layer M

$$\delta_j^{(M)} = (t_j - v_j^{(M)})(1 - v_j^{2(M)})/2 \quad (21)$$

and recursively back-propagating the error signal to lower layers

$$\delta_j^{(m)} = (1 - v_j^{2(m)}) \sum_l \delta_l^{(m+1)} w_{lj}^{(m+1)}/2 \quad (22)$$

where $m \in [1, 2, \ldots, M-1]$, l is over all neurons in the layer above neuron j and t_j is the desired output.

To allow rapid learning a momentum term, $\Delta w_{ij}^{(m)}(n)$, scaled by α is used to filter out high frequency variation of the weight vector. As a result, the convergence rate is much faster and the weight changes are smoothed.

6 Perceptron-based DFE performance and comparison with LMS DFE

The channel model used in the performance evaluation is given in z-transform notation by

$$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \quad (23)$$

The digital message applied to the channel was in random bipolar form $\{-1, 1\}$. The channel output is corrupted by zero mean white Gaussian noise. For mathematical convenience, we normalise the received signal power to unity. Then the received signal to noise ratio (SNR) is simply the reciprocal of the noise variance at the input of the equaliser.

The performance was determined by taking an average of 600 individual runs. Each run had a different random sequence and random starting weights. For simplicity the short hand notation {(N, l)DFE with (N_1, N_2, N_3)MLP} (MLP DFE) will be used to indicate that the number of received signal samples is N, the number of decision feedback samples is l, the number of neurons in hidden layer $1(H_1)$ is N_1, the number of neurons in hidden layer $2(H_2)$ is N_2, and the number of neurons in output layer is N_3, for a three layer perceptron based decision feedback equaliser.

6.1 Convergence characteristics

Fig. 8 illustrates MSE (mean square error) convergence of the MLP DFE, {(4, 1)DFE with (9, 3, 1)MLP structure}, with learning gain (η) 0.07 and the LMS DFE with learning gain (μ) 0.035. For illustrative purposes the MSE convergence of equalisers for no feedback signal (simple transversal equalisers) are shown. The MLP DFE requires at least 1000 iterations to converge while the LMS DFE converges in about 120 iterations. The results also show that the steady-state value of averaged square error produced by the MLP DFE converges to a value (< -25 dB) which is lower than the additive noise (-20 dB). This is a result of the nonlinear nature of equaliser transfer function. The LMS DFE gives a steady value of averaged square error at about -14.0 dB which is above the noise floor using the same number of input samples. The result also indicates that both types of the decision feedback equalisers yield a significant improvement in convergence time and averaged square error relative to the equalisers without feedback signal having the same number of input samples.

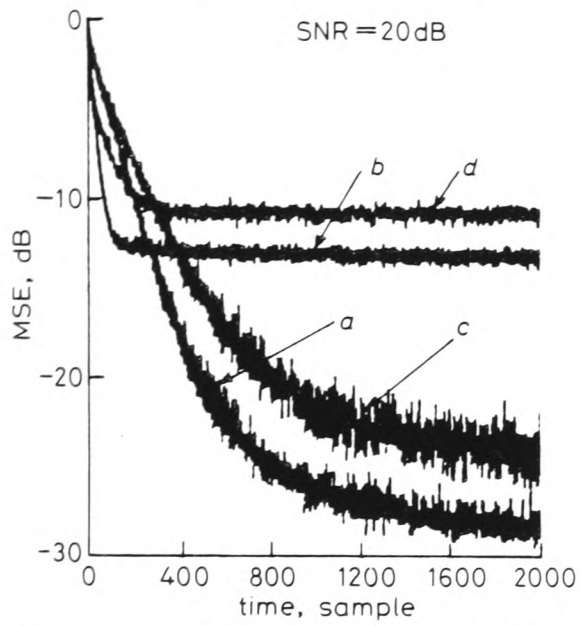


Fig. 8 *Simulation results showing relative convergence rate performance*

a (4, 1)DFE with (9, 3, 1)MLP structure: $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$
b LMS (4, 1)DFE structure: $\mu = 0.035$
c (5, 0)DFE with (9, 3, 1MLP structure (without feedback signal): $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$
d LMS (5, 0) DFE structure (without feedback signal): $\mu = 0.035$ at SNR = 20 dB

6.2 Decision region

Fig. 9 shows the decision region formed by a {(2, 0)DFE with (9, 3, 1)MLP structure} (without feedback signal) and the decision boundary formed by the optimal equaliser based on the *maximum a posteriori* (MAP) criterion [6], [8]. The signal to noise ratio was 10 dB. The

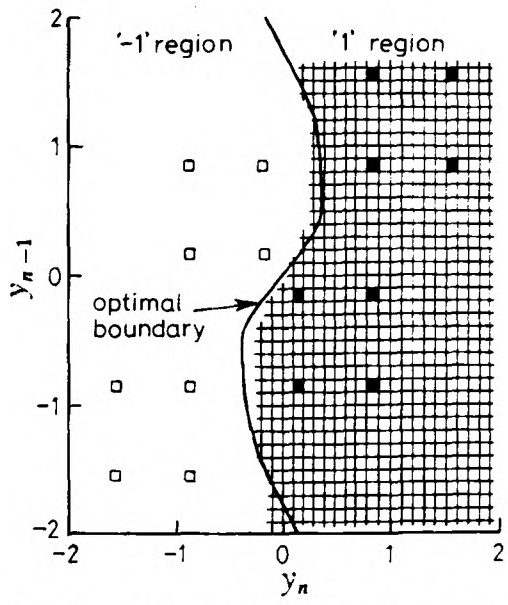


Fig. 9 *Decision region formed by (2, 0)DFE with (9, 3, 1)MLP structure after 300 samples: $\eta = 0.3$, $\alpha = 0.3$ and $\beta = 0.05$ where shading denotes decision region for '1' and optimal decision boundary formed by the MAP criterion with SNR = 10 dB*

Maximum a Posteriori criterion will yield a minimum probability-of-error decision. It can be seen that the decision region formed by the perceptron is near that optimal decision region which suggests that the perceptron is utilising the available information with something approaching maximum efficiency.

6.3 Bit error rate performance — decision directed mode

DFE performance can be obtained by means of a Monte Carlo simulation. Fig. 10 illustrates error rate performance in the stationary channel case for the MLP DFE performance using either correct or detected symbols in the feedback section with $\eta = 0.1$. For illustrative purposes the performance of the perceptron based equaliser (without feedback) with $\eta = 0.1$ is shown.

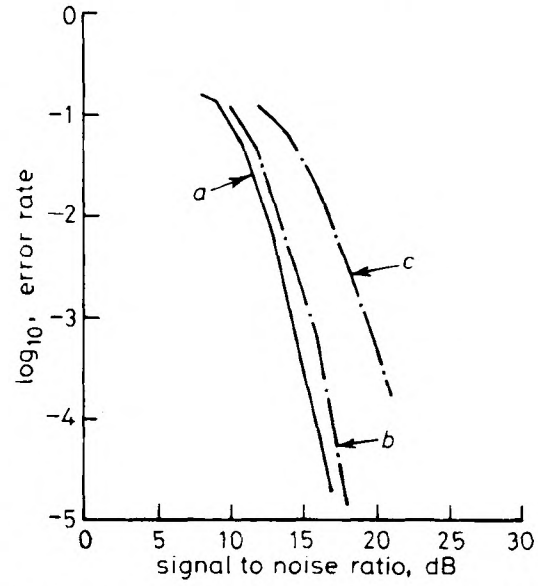


Fig. 10 *Performance of (4, 1)DFE with (9, 3, 1)MLP structure with and without error propagation, $\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$*

a Correct bit fed back
b Detected bit fed back
c (5, 0)DFE with (9, 3, 1)MLP structure (without feedback signal)

It may be observed from Fig. 10 that the MLP DFE attains about 5 dB improvement at BER = 10^{-4} relative to the MLP equaliser having the same number of input samples. The performance loss owing to incorrect deci-

sions being fed back is approximately 1.3 dB for the channel response under consideration.

Fig. 11 illustrates the performance of the LMS DFE with $\mu = 0.035$. The results show that the LMS DFE

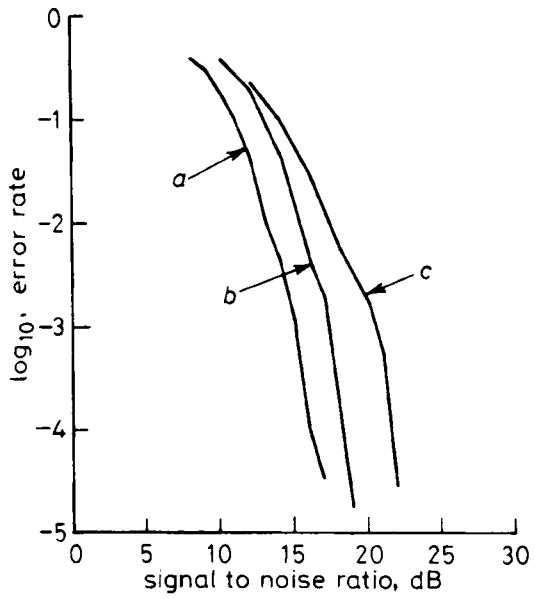


Fig. 11 *Performance of LMS (4, 1)DFE structure with and without error propagation, $\mu = 0.035$*

a Correct bit fed back
b Detected bit fed back
c LMS (5, 0)DFE structure (without feedback signal): $\mu = 0.035$

attains 5 dB improvement at BER = 10^{-4} relative to the LMS equaliser. The performance loss owing to incorrect decisions being fed back is about 2.0 dB. From the data in Figs. 10 and 11 it can be seen that the performance degradation owing to decision errors for the perceptron-based DFE is less than for the LMS DFE, especially under high noise conditions.

Fig. 12 illustrates performance for the MLP DFE with $\eta = 0.07$, 0.1 and the LMS DFE with $\mu = 0.035$, 0.05

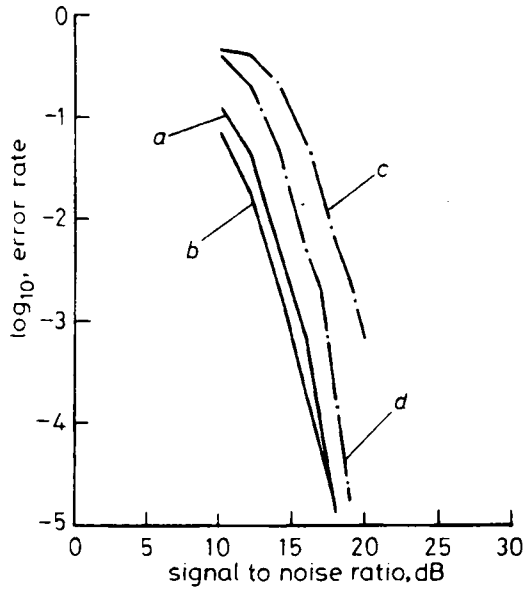


Fig. 12 *Simulation results showing relative BER performance for (4, 1) DFE with (9, 3, 1)MLP structure*

a $\eta = 0.1$, $\alpha = 0.3$, $\beta = 0.05$
b $\eta = 0.07$, $\alpha = 0.3$, $\beta = 0.05$

LMS (4, 1)DFE structure

c $\mu = 0.05$
d $\mu = 0.035$

having the same number of input samples. In the simulation, all the symbols fed back are detected symbols. The MLP DFE structure performs the superior performance in comparison with the LMS DFE structure, when the level of additive noise is high, but deteriorates as the signal to noise ratio improves. This latter fact should not surprise us since, if the additive noise level is very low, the MLP DFE will receive very few samples of signal which are close to the optimal decision boundary, rendering it incapable of forming optimal decision boundary as it does in the high noise situation.

7 Conclusions

The conventional structure of the DFE consists of a feed-forward equaliser and a feedback filter, where the feed-forward equaliser is linear. The linearity of the equaliser limits the performance of the system.

This paper has introduced a new approach for the DFE using multilayer structures. The back propagation learning algorithm is applied directly to the multilayer network. From comparison of simulation results it can be seen that the multilayer perceptron-based DFE provides better BER performance, especially in poor signal to noise ratio conditions, also that BER performance degrades less owing to decision errors and is also less sensitive to learning gain variation.

We conclude that the multilayer perceptron-based DFE offers a superior performance (higher resolution) as a channel equaliser to that of the conventional DFE, because of its ability to form complex decision regions with nonlinear boundaries. It should be noted, however, that the structures invoked here are considerably more complex than the conventional DFE. Further work aimed at comparisons with Viterbi based equalisers is currently being pursued.

8 Acknowledgments

The research was supported by the British Science and Engineering Research Council, the National Science Council of the Republic of China and Telecommunications Lab., Taiwan.

9 References

1 QURESHI, S.U.H.: 'Adaptive equalization', *Proc. IEEE*, 1985, **73**, (9), pp. 1349–1387
2 MINSKY, M., and PAPERT, S.: 'Perceptron: An introduction to computational geometry' (MIT Press, 1988)
3 RUMELHART, D.E., and McCLELLAND, J.L.: 'Parallel distributed processing: explorations in the microstructure of cognition' (MIT Press, 1986)
4 LIPPMANN, R.P.: 'An introduction to computing with neural nets', *IEEE ASSP Magazine*, April 1987, **4**, (2)
5 LAPEDES, A., and FARBER, R.: 'Nonlinear signal processing using neural networks: prediction and system modelling'. Los Alamos National Lab., Preprint La-Ur-87-2662
6 GIBSON, G.J., SIU, S., and COWAN, C.F.N.: 'Multi-layer perceptron structures applied to adaptive equalizers for data communications'. IEEE Proceedings ICASSP Glasgow, Scotland, May 1989, pp. 1183–1186
7 WIELAND, A., and LEIGHTON, R.: 'Geometric analysis of neural network capabilities'. 1st International Conference on Neural Networks, IEEE, June 1987, pp. 385–393
8 COUCH II, L.W.: 'Digital and analog communication systems' (Macmillan Publishing Co., Inc., 1983) Ch. 8, pp. 403–457
9 WIDROW, B., WINTER, R.G., and BAXTER, R.A.: 'Layered neural nets for pattern recognition', *IEEE Trans. ASSP*, July 1988, **36**, pp. 1096–1099